AD-A072 944 JOHNS HOPKINS UNIV LAUREL MD APPLIED PHYSICS LAB F/G 20/4

COMPUTATION OF WATER WAVES.(U)

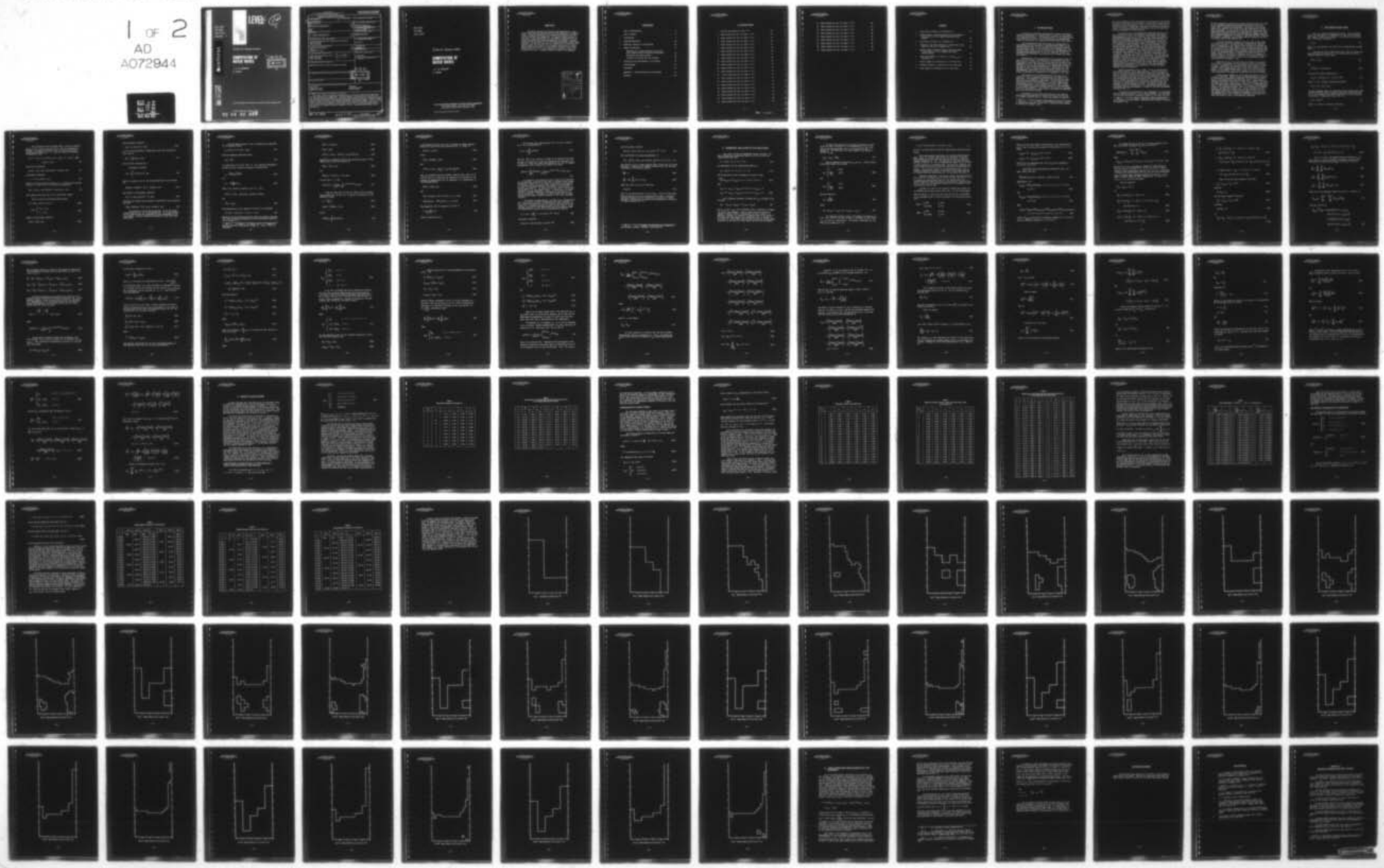
JUL 79 J C ROGERS, S FAVIN N00024-78-C-5384

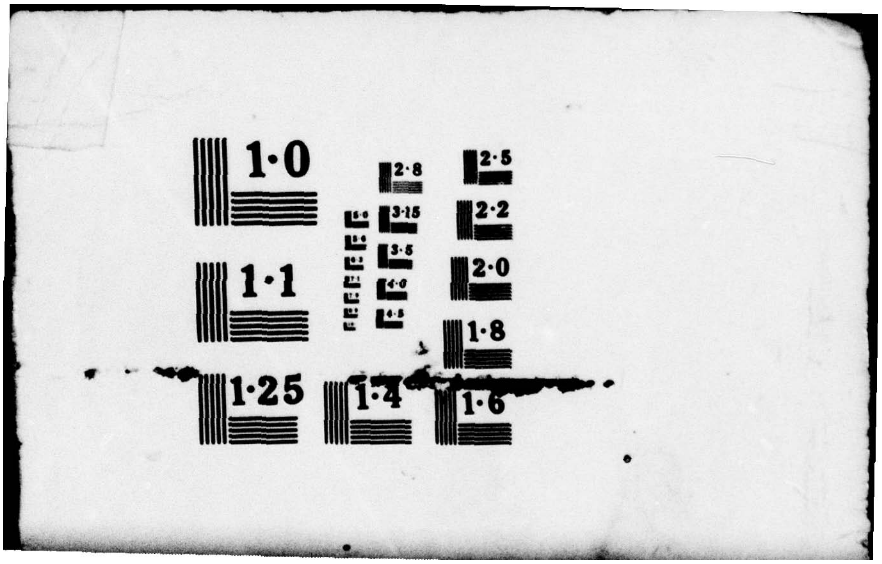
UNCLASSIFIED APL/JHU/TG-1325 NL

1 OF 2

AD

A072944




1·0
2·8
2·5
3·15
2·2
3·5
2·0
4·0
4·5
1·1
1·8
1·25
1·4
1·6

APL/JHU
TG 1325
JULY 1979
Copy No. 9

Technical Memorandum

COMPUTATION OF WATER WAVES

J. C. W. ROGERS
S. FAVIN

THE JOHNS HOPKINS UNIVERSITY ■ APPLIED PHYSICS LABORATORY

Approved for public release; distribution unlimited.

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

PLEASE FOLD BACK IF NOT NEEDED FOR BIBLIOGRAPHIC PURPOSES

REPORT DOCUMENTATION PAGE

14 ~~1. REPORT NUMBER~~
APL/JHU/TG-1325

2. GOVT ACCESSION NO

3. RECIPIENT'S CATALOG NUMBER

6 ~~4.~~ TITLE (and Subtitle)
COMPUTATION OF WATER WAVES

5. TYPE OF REPORT & PERIOD COVERED
9 Technical Memo~~randum~~

~~6. PERFORMING~~ ORG. REPORT NUMBER
APL/JHU TG 1325

10 ~~7. AUTHOR(s)~~
J.C.W. Rogers ~~and~~ S. Favin

8. CONTRACT OR GRANT NUMBER(s)
15 N00024-78-C-5384

9. PERFORMING ORGANIZATION NAME & ADDRESS
The Johns Hopkins University Applied Physics Laboratory
Johns Hopkins Rd.
Laurel, MD 20810

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
Task ZL50

11. CONTROLLING OFFICE NAME & ADDRESS
Office of Naval Research
Code 438
Arlington, VA 22217

12. REPORT DATE
11 July 1979

~~13. NUMBER OF~~ PAGES
131

14. MONITORING AGENCY NAME & ADDRESS
Naval Plant Representative Office
Johns Hopkins Rd.
Laurel, MD 20810

12 134p.

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Water waves	Turbulence
Conservation laws	Boltzmann equation
Numerical solution	Stefan problem

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The report contains the numerical implementation of a theoretical algorithm that generalizes the Euler equations for irregular flows. The main features of the algorithm are described, and the relevant equations are summarized. The spatial region occupied by the fluid is then discretized, and the numerical quadrature of the theoretical algorithm's governing equations is effected. Computational results are given for the following hydrodynamic free-boundary problems: (a) the fall from rest of a liquid with a profoundly nonlinear initial free surface; (b) the motion from rest of a liquid whose initial free surface is only slightly removed from its equilibrium position; and (c) the collisions of streams of fluid with formation of a jet. A program listing is given in the appendix.

031 650

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

JB

APL/JHU
TG 1325
JULY 1979

Technical Memorandum

COMPUTATION OF WATER WAVES

J. C. W. ROGERS
S. FAVIN

THE JOHNS HOPKINS UNIVERSITY ■ APPLIED PHYSICS LABORATORY
Johns Hopkins Road, Laurel, Maryland 20810
Operating under Contract N00024-78-C-5384 with the Department of the Navy

ABSTRACT

The report contains the numerical implementation of a theoretical algorithm that generalizes the Euler equations for irregular flows. The main features of the algorithm are described, and the relevant equations are summarized. The spatial region occupied by the fluid is then discretized, and the numerical quadrature of the theoretical algorithm's governing equations is effected. Computational results are given for the following hydrodynamic free-boundary problems: (a) the fall from rest of a liquid with a profoundly nonlinear initial free surface; (b) the motion from rest of a liquid whose initial free surface is only slightly removed from its equilibrium position; and (c) the collisions of streams of fluid with formation of a jet. A program listing is given in the appendix.

CONTENTS

ILLUSTRATIONS

PAGE 4 BLANK

TABLES

1. INTRODUCTION

As part of a continuing investigation, we have reformulated inviscid hydrodynamics in a manner that is natural for the study of the free-surface problem. The ultimate purpose of the investigation is to compute in an efficient and reliable way the phenomena attendant to the motion of a rigid body in water with a free surface.

The considerations that have guided our reformulation of hydrodynamics and the reduction of our generalized theory to the classical theory when the flow variables are sufficiently smooth were discussed in Ref. 1, and they need not be repeated here. The purpose of this report is to see directly what the implications of the generalized hydrodynamics are for the numerical solution of problems with hydrodynamic free surfaces and to present our numerical results.

Nevertheless, some recapitulation of the essential ideas of the theory is in order. Hydrodynamics as reformulated does not take on quite a Lagrangian or an Eulerian mode but has some of the advantages of both. The advantage our formulation shares with the Eulerian one is that the equations have as independent variables the ones of direct physical significance — space and time. The advantage it has in common with the Lagrangian approach is that time-dependent free-surface problems may be solved by solving a system of equations on a domain that is independent of the time.

The evolution of an incompressible flow is seen as the solution of a set of hyperbolic conservation laws subject to a constraint. The hyperbolic conservation laws are just the equations of mass and momentum conservation. In the absence of the constraint, they are the equations of a perfectly compressible (pressureless) fluid. (In N dimensions, we also refer to these as the N-dimensional inviscid Burgers equation.) The constraint is a one-sided constraint on the density that expresses incompressibility by establishing an upper bound on the amount of fluid that may occupy any volume of space.

In practice, in our theory the solution of the combined conservation laws/constraint is achieved by going from one time to a slightly enhanced one in a "split-step" scheme, in which at

Ref. 1. J. C. W. Rogers, "Incompressible Flows as a System of Conservation Laws with a Constraint," Seminaires IRIA, Analyse et Contrôle de Systèmes, 1978.

first the conservation laws are solved as if there were no constraint, and then the constraint is satisfied in a manner that retains the conservation of mass and momentum. Thus, if the solution of the evolutionary problem at a given time is thought of as obtained through the action of a nonlinear semigroup on the initial data, our theory provides an approximation to the semigroup.

In Section 2, we will summarize the governing equations that are solved at each time. Some numerical quadratures that enable these equations to be put in a finitary form suitable for computer manipulation are given in Section 3. Section 4 gives numerical results for three sample problems that have been run. The first of these follows the time evolution of an incompressible fluid that was initially at rest and whose initial free surface was distorted in a profoundly nonlinear manner. In the second problem, a comparison is made between the computer results and the predictions of linearized water wave theory. The third example illustrates the computation of phenomena associated with the collision of two streams and the formation of a jet. Here there is no exact theory to compare. Instead, the results of numerical computations are compared for different mesh sizes and time steps. All the numerical examples presented here are for problems with two independent space variables. Section 5 discusses some of the limitations of our method, possible improvements, and some of our plans for further computational work. In the belief that nothing removes ambiguity like an explicit statement of the steps we have gone through to implement our theoretical ideas numerically, we have included a program listing in Appendix A.

Several observations are in order. First, the generalized hydrodynamic theory is by no means completed. Since our theory is given in constructive fashion through an algorithm to determine the flow at any time in terms of the flow a time step earlier, the theory will be acceptable only when we have obtained definitive results for the construction as the time step goes to zero. Such results will have to include the regularity of the flow, convergence of the construction to a semigroup in the appropriate function spaces, and existence of the flow globally in time. Indications to date are that the conclusion of these tasks in a satisfactory manner will be concomitant with the development of a theory of inviscid hydrodynamic turbulence (Ref. 2).

In spite of this reservation, the commencement of calculations with the generalized theory is by no means premature. For one thing, the fact that the general formulation reduces to the usual one when

Ref. 2. J. C. W. Rogers, "Stability, Energy Conservation, and Turbulence for Water Waves," Seminaires IRIA, Analyse et Contrôle de Systèmes, 1978.

the flow variables are sufficiently smooth guarantees that the status of the fundamental questions mentioned above is no worse in our theory than in classical hydrodynamics. Accordingly, calculations based on our approach are no less sound than those based on conventional approaches. To the contrary, the fact that the general theory makes sense in a wider variety of circumstances and under less regularity requirements on the flow than the usual one suggests that perhaps a greater presumption of success in resolving the basic questions of existence, regularity, and turbulence may be attached to the general theory than to the classical one.

A second observation relates to the fact that in most cases of interest the flow of an unstratified, incompressible, inviscid fluid is irrotational. The general theory applies to rotational as well as irrotational flows. However, for the important special case of irrotational flows, significant savings in computational time may be effected in the classical theory through the use of integral equations. An unfinished task is to examine the ramifications of the general theory for initially irrotational flows and to develop, to whatever extent it is possible, variations on our algorithm that retain its essential character but effect the computational savings anticipated for this special case.

Therefore, we do not consider the algorithm we have studied to be final, by any means, for the majority of cases of practical interest. Accordingly, we have not expended great energy in trying to perfect numerically the algorithm in the form in which it now stands. From a computational point of view, this report should be viewed as a preliminary report whose purpose is to show the essential correctness of our approach, in practice as well as in theory.

We hope these comments put the accompanying numerical work in a proper perspective. While we have not been deliberately careless in carrying out the numerical quadratures, we have also not made a number of refinements that could have been made. (These points are discussed more fully in Section 5.) Thus, to some extent, the theory has been tested under rather adverse conditions. Our expectation is that, if the numerical implementation of the theory thus handicapped yields at all reasonable results, more can be expected of a similar approach, executed with greater care. We have ventured forth in this manner, guided by our faith that the theory, free as it is of the comparatively severe regularity requirements of the classical hydrodynamic theory, has an essential robustness that enables it to carry the weight of even a crude numerical quadrature.

2. GOVERNING EQUATIONS

Let D be a domain occupied by the fluid. In the problems treated in this paper, D is independent of time, and the boundary ∂D is rigid. D need not be bounded. ρ will denote the fluid density. The density constraint is

$$\rho \leq \rho_0, \tag{1}$$

where ρ_0 is the density of the fluid in its incompressible (liquid) phase.

The velocity field is denoted by u(x,t), and, of course, the momentum density is $\rho(x,t)u(x,t)$. The evolutionary problem is to find $\rho(x,t)$ and $\rho(x,t)u(x,t)$ given

$$\rho^0(x) = \rho(x,0) \tag{2a}$$

and

$$\rho^0(x)u^0(x) = \rho(x,0)u(x,0). \tag{2b}$$

We write the solution symbolically as

$$[\rho(x,t),\ \rho(x,t)u(x,t)] = S(t)\ (\rho^0,\ \rho^0u^0), \tag{3}$$

where S(t) is a nonlinear semigroup satisfying

$$S(t_1 + t_2) = S(t_1)\ S(t_2). \tag{4}$$

(In some turbulent flows, the system may evolve stochastically even though the initial state is uniquely prescribed, and in that case Eq. 4 is a statement of the Markov property of the time evolution of the flow.) A particular case of Eq. 4 is

$$S(t) = [S(\tau)]^n, \tag{5}$$

where $t = n\tau$ and τ is called the time step.

The algorithm we have introduced (Ref. 1) gives an approximation to $S(\tau)$, which we denote by $\overline{S}(\tau)$. $\overline{S}(\tau)$ is determined as follows: we suppose we are given $\rho(x)$ and $\rho(x)u(x)$, with $\rho(x)$ satisfying the constraint Eq. 1. First, we solve the hyperbolic "conservation" laws,

$$\xi_t + \nabla \cdot (\xi\eta) = 0,\ (x,t)\varepsilon D \times (0,\tau),\ (\xi\eta)_t + \nabla \cdot (\xi\eta\eta) = -\xi g\vec{k},$$

$$(x,t)\varepsilon D \times (0,\tau), \tag{6}$$

with initial conditions

$$\xi(x,0) = \rho(x),\ x\varepsilon D,\ \xi(x,0)\eta(x,0) = \rho(x)u(x),\ x\varepsilon D, \tag{7}$$

and boundary conditions

$$\eta \cdot n = 0,\ (x,t)\varepsilon\partial D \times (0,\tau), \tag{8}$$

where n is the unit outward normal to ∂D. In terms of the solution of Eqs. 6, 7, and 8 at $t = \tau$, we define the quantities

$$\tilde{\rho}(x) = \xi(x,\tau),\ x\varepsilon D,\ \tilde{\rho}(x)\tilde{u}(x) = \xi(x,\tau)\eta(x,\tau),\ x\varepsilon D.. \tag{9}$$

This completes the first part of our "split-step" scheme.

Next, we solve the one-phase Stefan problem

$$\theta_\alpha = \Delta f(\theta),\ (x,\alpha)\varepsilon D \times (0,\infty), \tag{10a}$$

$$f(\theta) = \begin{cases} \theta - \rho_0 & \theta \geq \rho_0 \\ 0 & \theta \leq \rho_0 \end{cases}, \tag{10b}$$

subject to the initial condition

$$\theta(x,0) = \tilde{\rho}(x),\ x\varepsilon D, \tag{10c}$$

and the boundary condition

$$\nabla\theta\cdot n = 0,\ (x,\alpha)\varepsilon\ \partial D \times (0,\infty). \qquad (10d)$$

As $\alpha\to\infty$, $\theta(x,\alpha)$ approaches a steady-state value that we denote by $\bar{\rho}(x)$; that is,

$$\bar{\rho}(x) \equiv \lim_{\alpha\to\infty} \theta(x,\alpha),\ x\varepsilon D. \qquad (11)$$

Note that $\bar{\rho}(x)$ satisfies Eq. 1.

Continuing, we define

$$v(x) \equiv \int_0^\infty f\ [\theta(x,\alpha)]\ d\alpha,\ x\varepsilon D, \qquad (12)$$

where θ is given by Eq. 10, and we determine $\bar{u}(x)$ as the solution of

$$\bar{\rho}(x)\bar{u}(x) = \tilde{\rho}(x)\tilde{u}(x) - \frac{2}{\tau}\nabla v + \Delta[v\bar{u}(x)],\ x\varepsilon D, \qquad (13a)$$

and subject to the boundary conditions

$$\bar{u}\cdot n = 0,\ x\varepsilon\partial D,\ nx(n\cdot\nabla)\bar{u} = 0,\ x\varepsilon\partial D. \qquad (13b)$$

With $\bar{\rho}(x)$ and $\bar{\rho}(x)\bar{u}(x)$ thus obtained, we define $\bar{S}(\tau)$ as the operator such that

$$[\bar{\rho}(x),\ \bar{\rho}(x)\bar{u}(x)] = \bar{S}(\tau)\ [\rho(x),\ \rho(x)u(x)],\ x\varepsilon D. \qquad (14)$$

The solution of the Stefan problem (Eq. 10) and the linear elliptic equation (Eq. 13) is straightforward. It is the solution of the Stefan problem that determines the time development of the free boundary. Explicit numerical algorithms to solve these problems follow.

For the Stefan problem, we use a variation on an algorithm for the general problem

$$u_t + Lf(u) = 0, \; u(x,0) = u_0(x), \tag{15}$$

when the semigroup associated with L,

$$S(t) = e^{-Lt}, \tag{16}$$

is contractive in L^1 and L^∞ (Ref. 3). The algorithm approximates $u(x,nh)$ by $u^n(x)$, obtained by making the substitutions

$$u_t \rightarrow \frac{u^n(x) - u^{n-1}(x)}{h} \tag{17a}$$

and

$$Lu \rightarrow - \frac{S(Ah) - 1}{Ah} u, \tag{17b}$$

where A is a positive constant, in Eq. 15. Thus,

$$u^{n+1}(x) = u^n(x) - \frac{1}{A} f[u^n(x)] + \frac{1}{A} S(hA) \; f[u^n(x)]$$

and

$$u^0(x) = u_0(x). \tag{18}$$

The algorithm (Eq. 18) is stable in L^1 and L^∞ if f satisfies

$$0 \leq f(u) - f(v) \leq A(u - v) \text{ for } u - v \geq 0. \tag{19}$$

Making the obvious substitutions and noting the boundary condition (Eq. 10d), we obtain an algorithm to solve the Stefan problem (Eq. 10) by setting A = 1:

Ref. 3. H. Brezis, A. E. Berger, and J. C. W. Rogers, "A Numerical Method for Solving the Problem $u_t - \Delta f(u) = 0$" (to be published).

$$\theta^n(x) \approx \theta(x, n\Delta\alpha), \tag{20a}$$

$$\theta^0(x) = \tilde{\rho}(x), \tag{20b}$$

$$\theta^{n+1}(x) = \theta^n(x) - f[\theta^n(x)] + S(\Delta\alpha)\ f[\overline{\theta}^n(x)], \tag{20c}$$

where $\overline{\theta}^n(x)$ is obtained from $\theta^n(x)$ by reflecting values of $\theta^n(x)$ symmetrically across the boundary ∂D:

$$\overline{\theta}^n(x) = \theta^n(x),\ x \varepsilon D, \tag{20d}$$

and

$$\overline{\theta}^n(x+n\varepsilon) = \theta^n(x-n\varepsilon),\ \varepsilon \geq 0,\ x\varepsilon\partial D. \tag{20e}$$

In this case, $L = -\Delta$ and

$$[S(h)u](x) = \frac{1}{(4\pi h)^{N/2}} \int_{R^N} e^{-(x-x')^2/4h}\ u(x')dx'. \tag{21}$$

There are many ways to solve the linear elliptic boundary value problem (Eq. 13) for $\overline{\rho}\ \overline{u}$. In terms of the solution of the parabolic problem

$$\psi_\gamma = \Delta(\frac{v}{\rho_0}\ \psi), \tag{22a}$$

$$\psi(x,0) = \tilde{\rho}(x)\tilde{u}(x) - \frac{2}{\tau}\ \nabla v, \tag{22b}$$

we get

$$\overline{\rho}(x)\overline{u}(x) = \int_0^\infty \psi(x,\gamma)e^{-\gamma}d\gamma. \tag{23}$$

An approximate solution of Eq. 22 is obtained by making substitutions in Eq. 22a of the sort indicated in Eq. 17. Thus, if

$$\psi^n(x) \approx \psi\ (x,n\Delta\gamma), \tag{24a}$$

we get

$$\psi^0(x) = \tilde{\rho}(x)\tilde{u}(x) - \frac{2}{\tau}\nabla v, \tag{24b}$$

and

$$\psi^{n+1}(x) = \psi^n(x) - \frac{1}{A}\frac{v}{\rho_0}\psi^n + \frac{1}{A}S(A\Delta\gamma)(\frac{v}{\rho_0}\overline{\psi}^n). \tag{24c}$$

Here, in accordance with the boundary conditions (Eq. 13b), we obtain $\overline{\psi}^n$ from ψ^n by reflecting the components of ψ^n parallel to the boundary ∂D symmetrically and the component of ψ^n perpendicular to ∂D antisymmetrically:

$$\overline{\psi}^n(x) = \psi^n(x),\ x\epsilon D, \tag{24d}$$

and

$$(\overline{\psi}^n \times n)(x+n\epsilon) = (\psi^n \times n)(x-n\epsilon),\ \epsilon \geq 0,\ x\epsilon\partial D, \tag{24e}$$

$$(\overline{\psi}^n\cdot n)(x+n\epsilon) = -(\overline{\psi}^n\cdot n)(x-n\epsilon),\ \epsilon \geq 0,\ x\epsilon\partial D. \tag{24f}$$

The scheme (Eq. 24c) is stable in L^1 and L^∞ if

$$A \geq \frac{1}{\rho_0} \sup_{x\epsilon D} v(x). \tag{25}$$

$S(A\Delta\gamma)$ is given by Eq. 21.

The function v(x), which appears in Eq. 24c and is defined in Eq. 12, is approximated by

$$v(x) \approx \Delta\alpha \sum_{n=0}^{\infty} f[\theta^n(x)]. \tag{26}$$

The term $-\frac{2}{\tau}\nabla v$ in Eq. 13a had its origin in the momentum associated with the redistribution of mass upon satisfying the constraint (Eq. 1) (Ref. 1). Thus, this term, which appears in Eq. 24b, is computed approximately from a formula that reflects its origin:

$$-\frac{2}{\tau}\nabla v(x) \approx \sum_{n=0}^{\infty} \int \frac{x - x'}{\tau} \, \frac{1}{(4\pi\Delta\alpha)^{N/2}} \, e^{-(x-x')^2/4\Delta\alpha} \, f[\theta^n(x')]dx'. \tag{27}$$

In contrast to the solution of Eqs. 10 and 13, the solution of the hyperbolic conservation laws (Eqs. 6 through 9) poses larger theoretical problems, as it is closely connected with the origins of turbulence and in the general case requires the enlargement of the class of acceptable solutions to stochastic flows in order to be well posed (Ref. 2). At this point we come close to the current limitations of our hydrodynamic theory. In particular, we do not now have a reliable algorithm to determine the evolution of the flow in probability. Accordingly, we shall assume that all flows studied in this report, whether turbulent or not, evolve deterministically.

The purpose in regarding Eqs. 6 through 9 as a "conservation" law is to provide a guide for determining the "weak" solution of Eq. 6 when the initial data $\rho(x)$ and $\rho(x)u(x)$ lack sufficient regularity for a classical solution to exist for all $t\epsilon(0,\tau)$. An approximate solution of this conservation law can be given in terms of a distribution function $F(x,v,t)$ satisfying the equation

$$F_t + v\cdot\nabla F - g\,\frac{\partial F}{\partial v_z} = 0, \quad (x,v,t)\epsilon D \times R^N \times(0,\tau), \tag{28}$$

the initial condition

$$F(x,v,0) = \rho(x)\ \delta[v-u(x)], \quad (x,v)\epsilon D \times R^N, \tag{29}$$

and the boundary condition

$$F(x,v,t) = F(x,v-2nv\cdot n,t),\quad (x,v,t)\epsilon\partial D \times R^N \times (0,\tau). \tag{30}$$

$\tilde{\rho}(x)$ and $\tilde{\rho}(x)\tilde{u}(x)$ are given approximately by

$$\tilde{\rho}(x) = \int F(x,v,\tau)dv,\ x\epsilon D,\ \tilde{\rho}(x)\tilde{u}(x) = \int F(x,v,\tau)v\ dv,\ x\epsilon D. \tag{31}$$

Note that Eq. 28 is a linear equation whose solution can be written explicitly. Because of the boundary condition (Eq. 30), the characteristics of the equation satisfy

$$\frac{dx}{dt} = v, \tag{32a}$$

$$\frac{dv}{dt} = -g\vec{k} - \sum_k 2nv\cdot n\ \delta(t-t_k), \tag{32b}$$

where the times $\{t_k\}$ are the times when

$$x(t_k)\epsilon\partial D \tag{32c}$$

and n is the outward normal to ∂D at $x(t_k)$. The use of the distribution function to solve the conservation law (Eqs. 6 through 9) has some similarity to the construction of solutions of another hyperbolic conservation law through the superposition of solutions of linear equations (Ref. 4).

Ref. 4. J. C. W. Rogers, "An Algorithm for a Hyperbolic Free Boundary Problem," APL/JHU TG 1309, May 1977.

3. NUMERICAL SOLUTION OF THE EQUATIONS

Let x and z be the two independent spatial variables. We shall indicate how the equations given in the last section are treated numerically when

$$D = \{(x,z) \mid z > 0,\ 0 < x < X\}. \tag{33}$$

We approximate D by the computational domain D_c:

$$D_c = \{(x,z) \mid 0 < x < X,\ 0 < z < Z\}, \tag{34}$$

and we partition D_c into rectangles by a grid of lines

$$\prod (x-x_{i+1/2})\ (z-z_{j+1/2}) = 0,\ 0 \leq i \leq I,\ 0 \leq j \leq J \tag{35a}$$

with

$$x_{1/2} = 0,\ x_{i-1/2} < x_{i+1/2} \text{ for } 1 \leq i \leq I,\ x_{I+1/2} = X,$$

$$z_{1/2} = 0,\ z_{j-1/2} < z_{j+1/2} \text{ for } 1 \leq j \leq J,\ z_{J+1/2} = Z. \tag{35b}$$

The fundamental dependent variables are m_{ij}, the mass in the rectangle

$$R_{ij} \equiv (x_{i-1/2}, x_{i+1/2}) \times (z_{j-1/2}, z_{j+1/2}), \tag{36}$$

and μ_{xij} (resp. μ_{zij}), the x-component (resp. z-component) of momentum in the same rectangle. Each of these quantities should be labeled by another index to distinguish the time at which it is measured. However, for purposes of simplicity and in the spirit of the analytical description of the algorithm in Eqs. 6 through 14, we do not carry this index along and only indicate how to find these quantities at one time from their values one time step beforehand.

We first show the effect of solving the hyperbolic conservation law (Eqs. 6 through 9) on the quantities m_{ij}, μ_{xij}, μ_{zij}. Here we use the approximate form of the algorithm in Eqs. 28 through 31. The first step is to take account of the effect of gravity on μ_{zij}:

$$\hat{\mu}_{zij} = \mu_{zij} - \tau g m_{ij}. \tag{37}$$

Next we compute the velocities u_{ij} and w_{ij}. Given an appropriate small number $\varepsilon>0$, we have

$$u_{ij} = \begin{cases} 0 & m_{ij}<\varepsilon \\ \dfrac{\mu_{xij}}{m_{ij}} & m_{ij}\geq\varepsilon \end{cases} \tag{38a}$$

and

$$w_{ij} = \begin{cases} 0 & m_{ij}<\varepsilon \\ \dfrac{\hat{\mu}_{zij}}{m_{ij}} & m_{ij}\geq\varepsilon \end{cases} . \tag{38b}$$

The mass density is

$$\rho_{ij} = \frac{m_{ij}}{\Delta x_i \Delta z_j} , \tag{39}$$

where

$$\Delta x_i = x_{i+1/2} - x_{i-1/2}, \ \Delta z_j = z_{j+1/2} - z_{j-1/2}. \tag{40}$$

Our numerical solution of Eqs. 28 through 31 proceeds as if $\rho(x)$, $u(x)$, and $w(x)$ are each constant in rectangle R_{ij} with values ρ_{ij}, u_{ij}, and w_{ij}, respectively. The boundary conditions (Eq. 30) hold on the rigid part of ∂D:

$$\Gamma \equiv \{0\} \times (0,Z) \bigcup (0,X) \times \{0\} \bigcup \{X\} \times (0,Z). \qquad (41)$$

On $\partial D_c - \Gamma$, we allow fluid to leave the computational region and never return.

With these boundary conditions, the problem becomes determinate. In Eq. 37, we have taken account of the effect of gravity. There remains to solve an equation like Eq. 28 with $g = 0$ and the boundary conditions (Eq. 30). The equations of the characteristics are Eqs. 32 with $g = 0$. For our computational domain D_c the solution of these equations is straightforward. One need only translate each point of each rectangle R_{ij} along its appropriate characteristic for a time τ, find its new location in the computational grid, and, in addition, ascertain the number of reversals of the normal component of velocity that have taken place at Γ, in accordance with Eqs. 32.

Different cases arise, according to whether the characteristic for a point of a rectangle R_{ij} has or has not been reflected at $z = 0$, or whether the characteristic has left the computational grid. A similar situation arises with regard to reflection of characteristics at $x = 0$ and $x = X$, in particular, whether the total number of such reflections is even or odd.

Let $\vec{x} \epsilon D_c$ be a point in the original computational domain and let $\vec{x}^*(\vec{x}) \epsilon D_c$ be its new location after time τ, if its characteristic does not leave D_c. We may map each such point $\vec{x}$ into a point $\bar{\vec{x}}(\vec{x})$ of the rectangle $(0,2X) \times (0,2Z)$ according to the following prescription.

$$\bar{x}(\vec{x}) = \begin{cases} x^*(\vec{x}) & N_1 \text{ even} \\ 2X - x^*(\vec{x}) & N_1 \text{ odd} \end{cases}, \qquad (42a)$$

$$\bar{z}(\vec{x}) = \begin{cases} z^*(\vec{x}) & N_2 \text{ even} \\ 2Z - z^*(\vec{x}) & N_2 \text{ odd} \end{cases}, \qquad (42b)$$

where N_1 is the total number of reflections of the characteristic at $x = 0$ and X, and N_2 is the number of reflections at $z = 0$. We define

$$x_{i+1/2} = 2X - x_{2I-i+1/2}, \quad I+1 \leqslant i \leqslant 2I, \tag{43a}$$

$$z_{j+1/2} = 2Z - z_{2J-j+1/2}, \quad J+1 \leqslant j \leqslant 2J. \tag{43b}$$

With Eq. 43, the definition (Eq. 36) of R_{ij} may be extended to $1 \leqslant i \leqslant 2I$, $1 \leqslant j \leqslant 2J$.

Points $\vec{y} \epsilon D_c$ whose characteristics lead after time τ to a point $\vec{x}^* \epsilon D_c$ will have

$$\vec{\bar{x}}(\vec{y}) \epsilon \{\vec{x}^*, (x^*, 2Z-z^*), (2X-x^*, z^*), (2X-x^*, 2Z-z^*)\}. \tag{44}$$

Equivalently, let

$$R_{ij;k\ell} \equiv \{x \epsilon R_{ij} | x^*(x) \epsilon R_{k\ell}\}, \quad 1 \leqslant i \leqslant I, \; 1 \leqslant j \leqslant J, \; 1 \leqslant k \leqslant I, \; 1 \leqslant \ell \leqslant J, \tag{45a}$$

and

$$\bar{R}_{ij;k\ell} \equiv \{x \epsilon R_{ij} | \bar{x}(x) \epsilon R_{k\ell}\}, \quad 1 \leqslant i \leqslant I, \; 1 \leqslant j \leqslant J, \; 1 \leqslant k \leqslant 2I, \; 1 \leqslant \ell \leqslant 2J. \tag{45b}$$

Then

$$R_{ij;k\ell} = \bar{R}_{ij;k\ell} \cup \bar{R}_{ij;2I+1-k,\ell} \cup \bar{R}_{ij;k,2J+1-\ell} \cup \bar{R}_{ij;2I+1-k,2J+1-\ell}, \quad 1 \leqslant i \leqslant I, \; 1 \leqslant j \leqslant J, \; 1 \leqslant k \leqslant I, \; 1 \leqslant \ell \leqslant J. \tag{46}$$

Fluid in $\bar{R}_{ij;k\ell}$ will have its x-velocity reversed if $I + 1 \leqslant k \leqslant 2I$ and its z-velocity reversed if $J + 1 \leqslant \ell \leqslant 2J$.

It follows from Eq. 32 with $g = 0$ and the assumption that u and w are constant throughout R_{ij} that we can write

$$\{\bar{x}(x)|x\epsilon R_{ij}\} = \bigcup_{\alpha=1}^{I(i,j)} \bigcup_{\beta=1}^{J(i,j)} \tilde{R}_{ij;\alpha\beta}, \tag{47a}$$

where

$$\tilde{R}_{ij;\alpha\beta} \equiv [\xi^{-}(i,j,\alpha), \xi^{+}(i,j,\alpha)] \times [\eta^{-}(i,j,\beta), \eta^{+}(i,j,\beta)], \tag{47b}$$

and $I(i,j)$ is 1 or 2, according to whether all characteristics from R_{ij} have been reflected at $x = 0$ and X an equal or unequal number of times, respectively, and $J(i,j)$ is 0, 1, or 2, according to whether all characteristics from R_{ij} have left the computational grid, or whether those remaining have been reflected at $z = 0$ an equal or unequal number of times, respectively. To find $I(i,j)$, $J(i,j)$, $\xi^{\pm}$, $\eta^{\pm}$, we do the following.

We first construct

$$z_{ij}^{\pm} = z_{j\pm1/2} + w_{ij}\,\tau. \tag{48}$$

If

$$z_{ij}^{+} \geq Z \text{ and } z_{ij}^{-} \geq Z\text{:} \quad J(i,j) = 0; \tag{49a}$$

$$z_{ij}^{+} \geq Z \text{ and } z_{ij}^{-} < Z\text{:} \quad J(i,j) = 1,\ \eta^{-}(i,j,1) = z_{ij}^{-},$$

$$\text{and } \eta^{+}(i,j,1) = Z; \tag{49b}$$

$$z_{ij}^{-} \leq -Z \text{ and } z_{ij}^{+} \leq -Z\text{:} \quad J(i,j) = 0; \tag{49c}$$

$$z_{ij}^{-} \leq -Z \text{ and } z_{ij}^{+} > -Z\text{:} \quad J(i,j) = 1,\ \eta^{-}(i,j,1) = Z,$$

$$\text{and } \eta^{+}(i,j,1) = 2Z + z_{ij}^{+}; \tag{49d}$$

$$0 < z^{+}_{ij} < Z \text{ and } z^{-}_{ij} \geq 0\text{:}\quad J(i,j) = 1,\ \eta^{-}(i,j,1) = z^{-}_{ij},$$
$$\text{and } \eta^{+}(i,j,1) = z^{+}_{ij}; \tag{49e}$$

$$0 < z^{+}_{ij} < Z \text{ and } z^{-}_{ij} < 0\text{:}\quad J(i,j) = 2,\ \eta^{-}(i,j,1)$$
$$= 0, \eta^{+}(i,j,1) = z^{+}_{ij},\ \eta^{-}(i,j,2) = 2Z + z^{-}_{ij}, \text{ and } \eta^{+}(i,j,2) = 2Z; \tag{49f}$$

$$-Z < z^{+}_{ij} \leq 0 \text{ and } -Z < z^{-}_{ij} < 0\text{:}\quad J(i,j) = 1,\ \eta^{-}(i,j,1)$$
$$= 2Z + z^{-}_{ij}, \text{ and } \eta^{+}(i,j,1) = 2Z + z^{+}_{ij}. \tag{49g}$$

Next we find the unique integer m^{-} such that

$$x^{-}_{ij} = x_{i-1/2} + u_{ij}\tau + 2m^{-}X \tag{50a}$$

satisfies

$$0 \leq x^{-}_{ij} < 2X \tag{50b}$$

and the unique integer m^{+} such that

$$x^{+}_{ij} = x_{i+1/2} + u_{ij}\tau + 2m^{+}X \tag{51a}$$

satisfies

$$0 \leq x^{+}_{ij} < 2X. \tag{51b}$$

If

$$x^{+}_{ij} > x^{-}_{ij}\text{:}\quad I(i,j) = 1,\ \xi^{-}(i,j,1) = x^{-}_{ij}, \text{ and } \xi^{+}(i,j,1) = x^{+}_{ij} \tag{52a}$$

$$x^+_{ij} \le x^-_{ij}: \quad I(i,j) = 2,\; \xi^-(i,j,1) = 0, \xi^+(i,j,1) = x^+_{ij},$$

$$\xi^-(i,j,2) = x^-_{ij}, \text{ and } \xi^+(i,j,2) = 2X. \tag{52b}$$

Since ρ, u, and w are assumed constant throughout R_{ij} for each i and j, we have for the mass and momentum associated with the points $y \epsilon D_c$ such that $\bar{x}(y) \epsilon R_{k\ell}$, $1 \le k \le 2I$, $1 \le \ell \le 2J$,

$$m^*_{k\ell} = \sum_{i=1}^{I} \sum_{j=1}^{J} |\bar{R}_{ij;k\ell}| \rho_{ij}, \tag{53a}$$

$$\mu^*_{xk\ell} = \sum_{i=1}^{I} \sum_{j=1}^{J} |\bar{R}_{ij;k\ell}| \rho_{ij}\, u_{ij}, \tag{53b}$$

$$\mu^*_{zk\ell} = \sum_{i=1}^{I} \sum_{j=1}^{J} |\bar{R}_{ij;k\ell}| \rho_{ij}\, w_{ij}, \tag{53c}$$

where $|A|$ is the (Lebesgue) measure of the set A. From Eq. 47,

$$|\bar{R}_{ij;k\ell}| = \sum_{\alpha=1}^{I(i,j)} \sum_{\beta=1}^{J(i,j)} |\tilde{R}_{ij;\alpha\beta} \bigcap R_{k\ell}|. \tag{54}$$

From Eqs. 47b and 36,

$$\begin{aligned} |\tilde{R}_{ij;\alpha\beta} \bigcap R_{k\ell}| = \max \Big\{ & \min[\xi^+(i,j,\alpha), x_{k+1/2}] \\ & - \max[\xi^-(i,j,\alpha), x_{k-1/2}], 0 \Big\} \\ \times \max \Big\{ & \min[\eta^+(i,j,\beta), z_{\ell+1/2}] \\ & - \max[\eta^-(i,j,\beta), z_{\ell-1/2}], 0 \Big\}. \end{aligned} \tag{55}$$

The approximate numerical solution of the hyperbolic conservation law is completed by setting, with the help of Eq. 46, for $1 \leqslant i \leqslant I$ and $1 \leqslant j \leqslant J$,

$$\tilde{m}_{ij} = m^*_{ij} + m^*_{2I+1-i,j} + m^*_{i,2J+1-j} + m^*_{2I+1-i,2J+1-j}, \tag{56a}$$

$$\tilde{\mu}_{xij} = \mu^*_{xij} - \mu^*_{x2I+1-i,j} + \mu^*_{xi,2J+1-j} - \mu^*_{x2I+1-i,2J+1-j}, \tag{56b}$$

$$\tilde{\mu}_{zij} = \mu^*_{zij} + \mu^*_{z2I+1-i,j} - \mu^*_{zi,2J+1-j} - \mu^*_{z2I+1-i,2J+1-j}. \tag{56c}$$

We must now transcribe the algorithms for solution of the one-phase Stefan problem and the elliptic problem (Eq. 13) to their numerical context. As we can see in Eqs. 20c and 24c, the numerical implementation of the algorithms requires an appropriate numerical representation of the operator S(h) in Eq. 21. First, observe that S(h) can be factored:

$$S(h) = e^{h \frac{\partial^2}{\partial z^2}} \; e^{h \frac{\partial^2}{\partial x^2}} \equiv S_z(h)\, S_x(h), \tag{57a}$$

where

$$[S_x(h)u](x) = \frac{1}{(4\pi h)^{1/2}} \int_{R^1} e^{-(x-x')^2/4h}\, u(x')dx'. \tag{57b}$$

To the level of accuracy we have been considering, functions operated on by S_x will be constant on each interval $(x_{i-1/2}, x_{i+1/2})$. Thus, denoting the characteristic function of a set E by $\chi(E)$, and letting

$$\chi_i = \chi[(x_{i-1/2}, x_{i+1/2})], \tag{58a}$$

we will want to replace Eq. 57b by

$$S_x(h)u = \sum_i u_i \, S_x(h)\chi_i, \tag{58b}$$

where u_i is the value of u associated with cell i. When $\frac{\Delta\alpha}{(\Delta x_i)^2}$ is sufficiently small, it is most convenient to approximate $S_x(\Delta\alpha)x_i$ by a function that is also constant on each interval $(x_{k-1/2}, x_{k+1/2})$ and, furthermore, that is zero when $|k - i| > 1$. Thus, we shall write approximately

$$S_x(\Delta\alpha)\chi_i = \Delta x_i \left(\frac{c_i^-}{\Delta x_{i-1}} \chi_{i-1} + \frac{c_i^0}{\Delta x_i} \chi_i + \frac{c_i^+}{\Delta x_{i+1}} \chi_{i+1} \right), \tag{59}$$

where $c_i^{\pm}$ and c_i^0 are to be found. We shall determine the coefficients by requiring that the first three moments of x be equal for the expressions on the left- and right-hand sides of Eq. 59. Referring to Eq. 57b, we easily calculate

$$\int S_x(\Delta\alpha) \; \chi_i dx = \Delta x_i, \tag{60a}$$

$$\int x \; S_x(\Delta\alpha) \; \chi_i dx = x_i \; \Delta x_i \; , \tag{60b}$$

$$\int x^2 \; S_x(\Delta\alpha) \; \chi_i dx = (x_i{}^2 + \frac{1}{12}(\Delta x_i)^2 + 2\Delta\alpha) \; \Delta x_i, \tag{60c}$$

where

$$x_i \equiv \frac{1}{2}(x_{i-1/2} + x_{i+1/2}). \tag{60d}$$

Equating Eqs. 60a through 60c with the corresponding moments of the right-hand side of Eq. 59, we obtain the equations

$$c_i^- + c_i^0 + c_i^+ = 1, \tag{61a}$$

$$c_i^- x_{i-1} + c_i^0 \ x_i + c_i^+ x_{i+1} = x_i, \tag{61b}$$

$$c_i^- \left[x_{i-1}^2 + \frac{1}{12}(\Delta x_{i-1})^2\right] + c_i^0 \left[x_i^2 + \frac{1}{12}(\Delta x_i)^2\right] + {}^{-}c_i^+ \left[x_{i+1}^2 + \frac{1}{12}(\Delta x_{i+1})^2\right]$$

$$= x_i^2 + \frac{1}{12}(\Delta x_i)^2 + 2\Delta\alpha. \tag{61c}$$

The final result is

$$c_i^- = 3\Delta\alpha\left[\Delta x_{i-1/2}(\Delta x_{i-1} + \Delta x_i + \Delta x_{i+1})\right]^{-1}, \tag{62a}$$

$$c_i^+ = 3\Delta\alpha\left[\Delta x_{i+1/2}(\Delta x_{i-1} + \Delta x_i + \Delta x_{i+1})\right]^{-1}, \tag{62b}$$

$$c_i^0 = 1 - c_i^- - c_i^+, \tag{62c}$$

where

$$\Delta x_{i+1/2} \equiv \frac{1}{2}(\Delta x_i + \Delta x_{i+1}). \tag{62d}$$

Thus, for a function $u = \sum_i u_i \chi_i$, it follows from Eqs. 58b and 59 that we can write

$$\int_{R_{kj}} S_x(\Delta\alpha) \, udxdz = \sum_i \tilde{P}_{ik} \, u_i \Delta z_j, \tag{63a}$$

where

$$\tilde{P}_{ik} = \begin{cases} c_i^-\Delta x_i & k = i - 1 \\ c_i^0\Delta x_i & k = i \\ c_i^+\Delta x_i & k = i + 1 \\ 0 & |k - i| > 1 \end{cases} \quad . \qquad (63b)$$

In Eq. 20c, we observe that $S_x(\Delta\alpha)$ operates on functions that have been continued symmetrically across the boundaries $x = 0$ and X. Thus, if one were to represent numerically the effect of $S_x(\Delta\alpha)$ at cell k on a function f that has the value f_i in cell i and is extended to a function $\overline{f}$ according to the prescription (Eqs. 20d and e), the result would be

$$\frac{1}{\Delta x_k} \sum_i \tilde{P}_{ik}\, \overline{f}_i = \frac{1}{\Delta x_k} \sum_{i=1}^{I} P_{ik} f_i , \qquad (64)$$

where

$$P_{ik} = \begin{cases} \tilde{P}_{ik} & 2 \le i \le I - 1 \text{ or } 2 \le k < I - 1 \\ (c_1^- + c_1^0)\Delta x_1 & i = k = 1 \\ (c_I^0 + c_I^+)\Delta x_I & i = k = I \end{cases} \quad . \qquad (65)$$

All the quantities in Eq. 65 are calculable through Eqs. 62 and 63b, with the convention that

$$\Delta x_0 = \Delta x_{1/2} = \Delta x_1 , \qquad (66a)$$

$$\Delta x_{I+1/2} = \Delta x_{I+1} = \Delta x_I . \qquad (66b)$$

Similar results hold for the approximation of the operator $S_z(\Delta\alpha)$. Let

$$z_j = \frac{1}{2}(z_{j-1/2} + z_{j+1/2}), \tag{67a}$$

$$\Delta Z_{j+1/2} = \frac{1}{2}(\Delta z_j + \Delta z_{j+1}), \tag{67b}$$

$$\Delta z_0 = \Delta z_{1/2} = \Delta z_1, \tag{67c}$$

$$\Delta z_{J+1/2} = \Delta z_{J+1} = \Delta z_J. \tag{67d}$$

Then the effect, evaluated in cell ℓ, of $S_z(\Delta\alpha)$ operating on a function f that has the value f_j in cell j and is extended symmetrically to a function $\overline{f}$ according to Eqs. 20d and 20e, is, for $\frac{\Delta\alpha}{(\Delta z_j)^2}$ sufficiently small,

$$\frac{1}{\Delta z_\ell} \sum_j \tilde{Q}_{j\ell}\, \overline{f}_j = \frac{1}{\Delta z_\ell} \sum_{j=1}^{J} Q_{j\ell} f_j, \tag{68}$$

where

$$Q_{j\ell} = \begin{cases} \tilde{Q}_{j\ell} & 2 \le j \le J \text{ or } 2 \le \ell \le J \\ (e_1^- + e_1^0)\Delta z_1 & j = \ell = 1 \end{cases}, \tag{69}$$

$$\tilde{Q}_{j\ell} = \begin{cases} e_j^- \Delta z_j & \ell = j - 1 \\ e_j^0 \Delta z_j & \ell = j \\ e_j^+ \Delta z_j & \ell = j + 1 \\ 0 & |\ell - j| > 1 \end{cases} , \tag{70}$$

and

$$e_j^- = 3\Delta\alpha\left[\Delta z_{j-1/2}(\Delta z_{j-1} + \Delta z_j + \Delta z_{j+1})\right]^{-1}, \tag{71a}$$

$$e_j^+ = 3\Delta\alpha\left[\Delta z_{j+1/2}(\Delta z_{j-1} + \Delta z_j + \Delta z_{j+1})\right]^{-1}, \tag{71b}$$

$$e_j^0 = 1 - e_j^- - e_j^+. \tag{71c}$$

When $\Delta\alpha$ is not small enough, some of the quantities P_{ii} or Q_{jj} given by Eqs. 65 and 69 may be negative, and in that case the approximate representations of $S_x(\Delta\alpha)$ or $S_z(\Delta\alpha)$ that have been given above will lead to instabilities in the computation.

Accordingly, if, for example, $P_{ii} < 0$, it will be necessary for us to obtain a better representation of $S_x(\Delta\alpha)\chi_i$ than that afforded by Eq. 59. Referring to Eq. 57b, we look at

$$S_x(\Delta\alpha)\chi_i = \frac{1}{(4\pi\Delta\alpha)^{1/2}} \int_{x-x_{i+1/2}}^{x-x_{i-1/2}} e^{-\xi^2/4\Delta\alpha} d\xi, \tag{72}$$

when x is in the kth cell. Consistent with the accuracy of the numerical quadrature we have been using, we replace $\tilde{P}_{ik}$ in Eq. 63 by the integral of Eq. 72 over the kth cell. For $k > i$, this is

$$\tilde{P}_{ik} = \frac{1}{\sqrt{4\pi\Delta\alpha}} \int_{x_{k-1/2}}^{x_{k+1/2}} \int_{x-x_{i+1/2}}^{x-x_{i-1/2}} e^{-\xi^2/4\Delta\alpha}\, d\xi\, dx$$

$$= P\left(\frac{x_{k-1/2} - x_{i+1/2}}{2\sqrt{\Delta\alpha}}\right) - P\left(\frac{x_{k+1/2} - x_{i+1/2}}{2\sqrt{\Delta\alpha}}\right)$$

$$- P\left(\frac{x_{k-1/2} - x_{i-1/2}}{2\sqrt{\Delta\alpha}}\right) + P\left(\frac{x_{k+1/2} - x_{i-1/2}}{2\sqrt{\Delta\alpha}}\right) , \qquad (73a)$$

where

$$P(\xi) = \sqrt{\frac{\Delta\alpha}{\pi}} \left(e^{-\xi^2} - 2\xi \int_{\xi}^{\infty} e^{-\eta^2}\, d\eta\right) . \qquad (73b)$$

When k < i, one obtains

$$\tilde{P}_{ik} = \tilde{P}_{ki} . \qquad (74)$$

If $S_x(\Delta\alpha)$ operates on a function that has been continued symmetrically across the boundaries x = 0 and X, the result may be put in the form (Eq. 64) where now P_{ik} is given approximately by

$$
\begin{aligned}
P_{ik} = {} & P\left(\frac{x_{k-1/2} - x_{i+1/2}}{2\sqrt{\Delta\alpha}}\right) - P\left(\frac{x_{k+1/2} - x_{i+1/2}}{2\sqrt{\Delta\alpha}}\right) \\
& - P\left(\frac{x_{k-1/2} - x_{i-1/2}}{2\sqrt{\Delta\alpha}}\right) + P\left(\frac{x_{k+1/2} - x_{i-1/2}}{2\sqrt{\Delta\alpha}}\right) \\
& + P\left(\frac{x_{i-1/2} + x_{k-1/2}}{2\sqrt{\Delta\alpha}}\right) - P\left(\frac{x_{i-1/2} + x_{k+1/2}}{2\sqrt{\Delta\alpha}}\right) \\
& - P\left(\frac{x_{i+1/2} + x_{k-1/2}}{2\sqrt{\Delta\alpha}}\right) + P\left(\frac{x_{i+1/2} + x_{k+1/2}}{2\sqrt{\Delta\alpha}}\right) \\
& + P\left(\frac{2X - x_{k+1/2} - x_{i+1/2}}{2\sqrt{\Delta\alpha}}\right) - P\left(\frac{2X - x_{k-1/2} - x_{i+1/2}}{2\sqrt{\Delta\alpha}}\right) \\
& - P\left(\frac{2X - x_{k+1/2} - x_{i-1/2}}{2\sqrt{\Delta\alpha}}\right) + P\left(\frac{2X - x_{k-1/2} - x_{i-1/2}}{2\sqrt{\Delta\alpha}}\right),
\end{aligned}
$$

$$1 \leq i < k \leq I, \tag{75a}$$

$$P_{ik} = P_{ki}, \; 1 \leq k < i \leq I, \tag{75b}$$

$$P_{ii} = \Delta x_i - \sum_{\substack{k \neq i \\ 1 \leq k \leq I}} P_{ik}, \; 1 \leq i \leq I. \tag{75c}$$

Similarly, if Q_{jj} calculated by Eqs. 69 through 71 is < 0, we calculate $Q_{j\ell}$ according to the following procedure. Let

$$\mathcal{Q}_{j\ell} \equiv \frac{1}{\sqrt{4\pi\Delta\alpha}} \int_{z_{\ell-1/2}}^{z_{\ell+1/2}} \int_{z - z_{j+1/2}}^{z - z_{j-1/2}} e^{-\xi^2/4\Delta\alpha} \, d\xi \, dz. \tag{76}$$

When $j \neq \ell$, $\mathcal{Q}_{j\ell}$ is given by equations similar to Eqs. 73 and 74. For $\ell = j$, we find

$$\mathcal{Q}_{jj} = \Delta z_j - 2 \frac{\sqrt{\Delta\alpha}}{\pi} + 2P\left(\frac{\Delta z_j}{2\sqrt{\Delta\alpha}}\right) . \tag{77}$$

The effect of $S_z(\Delta\alpha)$ evaluated at cell ℓ operating on a function that has the value f_j in cell j and has been extended symmetrically across z = 0 according to Eqs. 20d and 20e may be put in the form (Eq. 68), where $Q_{j\ell}$ is given approximately by

$$\begin{aligned} Q_{j\ell} = {} & P\left(\frac{z_{\ell-1/2} - z_{j+1/2}}{2\sqrt{\Delta\alpha}}\right) - P\left(\frac{z_{\ell+1/2} - z_{j+1/2}}{2\sqrt{\Delta\alpha}}\right) \\ & - P\left(\frac{z_{\ell-1/2} - z_{j-1/2}}{2\sqrt{\Delta\alpha}}\right) + P\left(\frac{z_{\ell+1/2} - z_{j}{}^{-1/2}}{2\sqrt{\Delta\alpha}}\right) \\ & + P\left(\frac{z_{\ell-1/2} + z_{j-1/2}}{2\sqrt{\Delta\alpha}}\right) - P\left(\frac{z_{\ell+1/2} + z_{j-1/2}}{2\sqrt{\Delta\alpha}}\right) \\ & - P\left(\frac{z_{\ell-1/2} + z_{j+1/2}}{2\sqrt{\Delta\alpha}}\right) + P\left(\frac{z_{\ell+1/2} + z_{j+1/2}}{2\sqrt{\Delta\alpha}}\right), \\ & 1 \le j < \ell \le J, \end{aligned} \tag{78a}$$

$$Q_{j\ell} = Q_{\ell j}, \; 1 \leqslant \ell < j \leqslant J, \tag{78b}$$

$$Q_{jj} = \Delta z_j - 2\sqrt{\frac{\Delta\alpha}{\pi}} + 2P\left(\frac{\Delta z_j}{2\sqrt{\Delta\alpha}}\right) + P\left(\frac{z_{j-1/2}}{\sqrt{\Delta\alpha}}\right) - 2P\left(\frac{z_j}{\sqrt{\Delta\alpha}}\right)$$
$$+ P\left(\frac{z_{j+1/2}}{\sqrt{\Delta\alpha}}\right), \qquad 1 \leqslant j \leqslant J. \tag{78c}$$

In the numerical solution of the Stefan problem, we follow the prescription laid out in Eq. 20. We form quantities m^n_{ij}, starting with

$$m^0_{ij} = \tilde{m}_{ij}. \tag{79}$$

The step corresponding to Eq. 20c in which m^{n+1}_{ij} is computed is performed in two parts.

First, we compute

$$\rho^n_{ij} = \frac{m^n_{ij}}{\Delta x_i \Delta z_j} \tag{80}$$

and, given a small positive constant ε_1, we check whether or not

$$\max_{\substack{1 \leqslant i \leqslant I \\ 1 \leqslant j \leqslant J}} (\rho^n_{ij} - \rho_0) \leqslant \varepsilon_1. \tag{81}$$

The constant ε_1 is what terminates the solution of the Stefan problem and represents an allowable margin of error in the density calculation. Suppose Eq. 81 is first satisfied for $n = n_0$. Then we set

$$\bar{m}_{ij} = m_{ij}^{n_0}. \tag{82}$$

For $n < n_0$, we form

$$m_{ij}^{n+1/2} = \Delta x_i \Delta z_j \left[\rho_{ij}^{n} - f\left(\rho_{ij}^{n}\right)\right] + \Delta z_j \sum_{k=1}^{I} P_{ki}\, f\left(\rho_{kj}^{n}\right). \tag{83}$$

Then we compute

$$\rho_{ij}^{n+1/2} = \frac{m_{ij}^{n+1/2}}{\Delta x_i \Delta z_j} \tag{84}$$

and let

$$m_{ij}^{n+1} = \Delta x_i \Delta z_j \left[\rho_{ij}^{n+1/2} - f\left(\rho_{ij}^{n+1/2}\right)\right] + \Delta x_i \sum_{\ell=1}^{J} Q_{\ell j}\, f\left(\rho_{i\ell}^{n+1/2}\right). \tag{85}$$

In place of Eq. 26 we have

$$v_{ij} = \sum_{n=0}^{n_0-1} \Delta\alpha\; f(\rho_{ij}^{n}), \tag{86}$$

and Eq. 27 is the basis for the numerical analogs

$$(\Delta\mu)_{xij} = \Delta z_j \sum_{n=0}^{n_0-1} \sum_{k=1}^{I} P_{ki} f(\rho_{kj}^n)$$

$$\times \frac{1}{2}\left(x_{i-1/2} + x_{i+1/2} - x_{k-1/2} - x_{k+1/2}\right) \tag{87a}$$

and

$$(\Delta\mu)_{zij} = \Delta x_i \sum_{n=0}^{n_0-1} \sum_{\ell=1}^{J} Q_{\ell j} f\left(\rho_{i\ell}^{n+1/2}\right)$$

$$\times \frac{1}{2}\left(z_{j-1/2} + z_{j+1/2} - z_{\ell-1/2} - z_{\ell+1/2}\right) . \tag{87b}$$

Finally we determine the new momenta by using Eq. 24. We calculate quantities $\mu_{x_{ij}}^{p}$ and $\mu_{z_{ij}}^{p}$, starting with

$$\mu_{xij}^{0} = \tilde{\mu}_{xij} + \frac{1}{\tau} (\Delta\mu)_{xij}$$

and

$$\mu_{zij}^{0} = \tilde{\mu}_{zij} + \frac{1}{\tau} (\Delta\mu)_{zij}. \tag{88}$$

If

$$\max_{\substack{1 \leqslant i \leqslant I \\ 1 \leqslant j \leqslant J}} v_{ij} \leqslant \varepsilon_2 \tag{89}$$

where ε_2 is a small positive constant, we set

$$\bar{\mu}_{xij} = \mu^{0}_{xij}$$

and

$$\bar{\mu}_{zij} = \mu^{0}_{zij}. \tag{90}$$

Otherwise, let

$$v^{+} = \max_{\substack{1 \le i \le I \\ 1 \le j \le J}} v_{ij}. \tag{91}$$

Mindful of the sufficient condition for stability of the algorithm Eq. 24 as given in Eq. 25, we set

$$A = 1.1 \frac{v^{+}}{\rho_0}, \tag{92}$$

$$\Delta\gamma = \Delta\alpha/A, \tag{93}$$

and

$$v^{(1)}_{ij} = \frac{v_{ij}}{1.1v^{+}}. \tag{94}$$

Then we proceed with the algorithm Eq. 24 with this A and $\Delta\gamma$, computing μ^{p}_{xij} and μ^{p}_{zij} for $0 \le p \le p_0$ where p_0 is the first integer for which

$$(p_0 + 1)\, \Delta\gamma > \gamma_0, \tag{95}$$

and γ_0 is a prescribed positive constant with $e^{-\gamma_0}$ considered to be a small number.

The numerical step corresponding to Eq. 24c, in which μ^{p+1}_{xij} and μ^{p+1}_{zij} are found, is done in two parts. First, we find

$$\mu^{p+1/2}_{xij} = \sum_{\ell=1}^{J} \frac{Q_{\ell j}\ v^{(1)}_{i\ell}\ \mu^{p}_{xi\ell}}{\Delta x_i\ \Delta z_\ell} \tag{96a}$$

and

$$\mu^{p+1/2}_{zij} = \sum_{\ell=1}^{J} \frac{Q^{(1)}_{\ell j}\ v^{(1)}_{i\ell}\ \mu^{p}_{zi\ell}}{\Delta x_i\ \Delta z_\ell} \tag{96b}$$

Then we compute

$$\mu^{p+1/2}_{xij} = (1 - v^{(1)}_{ij})\ \mu^{p}_{xij} + \sum_{k=1}^{I} P^{(1)}_{ki}\ \mu^{p+1/2}_{xkj} \tag{97a}$$

and

$$\mu^{p+1}_{zij} = (1 - v^{(1)}_{ij})\ \mu^{p}_{zij} + \sum_{k=1}^{I} P_{ki}\ \mu^{p+1/2}_{zkj}. \tag{97b}$$

Here $P^{(1)}$ and $Q^{(1)}$ are found in a manner reminiscent of the derivation of P and Q above. The only difference is brought about by the different way $\bar{\psi}^n$ is obtained from ψ^n in Eq. 24 as compared with the extension of θ^n to $\bar{\theta}^n$ in Eq. 20. Specifically, when P_{ii} as calculated by Eqs. 62, 63, and 65 is ≥ 0,

$$P_{ik}^{(1)} = \begin{cases} P_{ik} & 2 \le i \le I - 1 \text{ or } 2 \le k \le I - 1 \\ (c_1^0 - c_1^-)\Delta x_1 & i = k = 1 \\ (c_I^0 - c_I^+)\Delta x_I & i = k = I \end{cases} , \quad (98)$$

and when Q_{jj} calculated by Eqs. 69 through 71 is ≥ 0,

$$Q_{j\ell}^{(1)} = \begin{cases} Q_{j\ell} & 2 \le j \le J \text{ or } 2 \le \ell \le J \\ (e_1^0 - e_1^-)\Delta z_1 & j = \ell = 1 \end{cases} . \quad (99)$$

If, on the other hand, Eqs. 62, 63, and 65 yield a value of $P_{ii} < 0$, $P_{ik}^{(1)}$ is given by

$$P_{ik}^{(1)} = 2P\left(\frac{x_{k-1/2} - x_{i+1/2}}{2\sqrt{\Delta\alpha}}\right) -2P\left(\frac{x_{k+1/2} - x_{i+1/2}}{2\sqrt{\Delta\alpha}}\right) -2P\left(\frac{x_{k-1/2} - x_{i-1/2}}{2\sqrt{\Delta\alpha}}\right)$$

$$+2P\left(\frac{x_{k+1/2} - x_{i-1/2}}{2\sqrt{\Delta\alpha}}\right) - P_{ik}, \quad 1 \le i < k \le I, \quad (100a)$$

$$P_{ik}^{(1)} = P_{ki}^{(1)} \quad , \quad 1 \le k < i \le I , \quad (100b)$$

$$P_{ii}^{(1)} = 2P\left(\frac{\Delta x_i}{2\sqrt{\Delta\alpha}}\right) + \Delta x_i - 2\sqrt{\frac{\Delta\alpha}{\pi}} - P\left(\frac{x_{i-1/2}}{\sqrt{\Delta\alpha}}\right) + 2P\left(\frac{x_i}{\sqrt{\Delta\alpha}}\right) - P\left(\frac{x_{i+1/2}}{\sqrt{\Delta\alpha}}\right)$$

$$- P\left(\frac{X - x_{i+1/2}}{\sqrt{\Delta\alpha}}\right) + 2P\left(\frac{X - x_i}{\sqrt{\Delta\alpha}}\right) - P\left(\frac{X - x_{i-1/2}}{\sqrt{\Delta\alpha}}\right)$$

$$1 \le i \le I. \tag{100c}$$

And if Eqs. 69 through 71 give $Q_{jj} < 0$, we calculate $Q_{j\ell}^{(1)}$ from the alternative formula

$$Q_{j\ell}^{(1)} = Q_{j\ell} - 2P\left(\frac{z_{j-1/2} + z_{\ell-1/2}}{2\sqrt{\Delta\alpha}}\right) + 2P\left(\frac{z_{j-1/2} + z_{\ell+1/2}}{2\sqrt{\Delta\alpha}}\right)$$

$$+ 2P\left(\frac{z_{j+1/2} + z_{\ell-1/2}}{2\sqrt{\Delta\alpha}}\right) - 2P\left(\frac{z_{j+1/2} + z_{\ell+1/2}}{2\sqrt{\Delta\alpha}}\right),$$

$$1 \le j \le J,\ 1 \le \ell \le J,\ j \ne \ell, \tag{101a}$$

$$Q_{jj}^{(1)} = \Delta z_j - 2\sqrt{\frac{\Delta\alpha}{\pi}} + 2P\left(\frac{\Delta z_j}{2\sqrt{\Delta\alpha}}\right) - P\left(\frac{z_{j-1/2}}{\sqrt{\Delta\alpha}}\right) + 2P\left(\frac{z_j}{\sqrt{\Delta\alpha}}\right)$$

$$- P\left(\frac{z_{j+1/2}}{\sqrt{\Delta\alpha}}\right), \quad 1 \le j \le J. \tag{101b}$$

Finally, the numerical version of Eq. 23 is

$$\bar{\mu}_{ij} = \sum_{p=0}^{P_0-1} \mu_{ij}^{p}\, e^{-p\Delta\gamma}\,(1 - e^{-\Delta\gamma}) + \mu_{ij}^{P_0}\, e^{-P_0\Delta\gamma}. \tag{102}$$

4. SAMPLE CALCULATIONS

We shall describe some calculations that we have made, using the numerical representation of the flow described in Section 3. A listing of the computer program appears in Appendix A. We shall outline the results of the calculations one by one and examine them for any indications they offer about the accuracy of the code. More detailed observations about the deficiencies of the present program and suggested improvements will be given in Section 5.

In each of the calculations, we have found it worthwhile to monitor the total energy as a function of time. In general, energy will be lost in the part of the algorithm that approximately solves the hyperbolic conservation laws (Eq. 6). This follows from Eq. 31, which expresses the fact that all "collisions" of parcels of fluid are inelastic. This energy loss is not just a function of the time and space discretization but may persist even in the continuous limit and, in fact, is intimately related to the turbulent or nonturbulent character of the flow (Ref. 2). Nevertheless, classical inviscid flows conserve energy, and by examining the variation of energy with the time we may get an idea either of the degree to which the flow is turbulent or of the error involved in the discrete approximation of the flow. The energy loss due to the discrete approximation essentially has two sources: a "diffusive" energy loss associated with the finite size of Δx and Δz, and a "collisional" loss associated with the finiteness of the time step τ and the possibility of different characteristics of the Boltzmann equation (Eq. 28) running together in time τ.

(The algorithm that we have described is not guaranteed to dissipate energy, since energy may be created in the solution of the Stefan problem (Eq. 10) and the elliptic boundary value problem (Eq. 13). However, one can verify that, for the analytical algorithm of Section 2, such energy production does not exceed the collisional loss in the limit as $\tau \to 0$ (Ref. 1). Thus, any energy increase that takes place in the numerical solution of the discrete equations must reflect the discretization error involved.)

EVOLUTION OF A LIQUID INITIALLY AT REST AND WITH A HIGHLY DISTORTED INITIAL FREE SURFACE

Our first calculation had $I = J = 10$, $\Delta x_i = \Delta z_j = 1$, $\tau = 0.1$, $g = 1$, and $\rho_0 = 1$. The initial data were

$$m_{ij} = \begin{cases} 1 & 1 \leqslant i \leqslant 2,\ 1 \leqslant j \leqslant 2 \\ 1 & 3 \leqslant i \leqslant 4,\ 1 \leqslant j \leqslant 10 \\ 1 & 5 \leqslant i \leqslant 7,\ 1 \leqslant j \leqslant 5 \\ 1 & 8 \leqslant i \leqslant 10,\ 1 \leqslant j \leqslant 3 \\ 0 & \text{otherwise,} \end{cases} \qquad (103)$$

and $\mu_{xij} = \mu_{zij} = 0$, $1 \leqslant i, j \leqslant 10$. Other parameters for the calculation were chosen as $\varepsilon = 10^{-5}$, $\varepsilon_1 = 10^{-2}$, $\varepsilon_2 = 10^{-3}$, $\gamma_0 = 10$, and $\Delta\alpha = 0.1$, as defined in Eqs. 38, 67, 75, 81, and 26, respectively. The calculation was run until time $t = 14$.

Our original hope in choosing the initial liquid domain as given in Eq. 103 was that such a highly distorted initial surface would lead to wave breaking and falling over, with the attendant formation of cavities. However, this expectation was not borne out by the computational output. The numerical results indicated a free surface for which the vertical coordinate was a single-valued function of the horizontal coordinate. In this case the z-coordinate can be identified with the total mass in a column of fluid. Table 1 gives the results for this total mass at times $t = 0$, 2, 4, 6, and 8. We observe that the free surface appears to oscillate in time, with the oscillations getting progressively smaller as time increases. There develops a rather high peak of the free surface at the side walls ($i = 1$ and 10), a feature that has dubious authenticity. Rather, this appears to be related to a defect of the numerical algorithm of Section 3 with regard to the treatment of normal components of velocities at rigid boundaries. This point is discussed more fully in Section 5, where we also suggest an improvement.

In Table 2, we show the kinetic energy (KE), potential energy (PE), and total energy (E) of the fluid system as a function of time. We note that there is an initial increase in energy. Of course, this is a spurious effect and indicates the size of the discretization error made. As time progresses, the energy appears to decay. We have pointed out above that, for a nonturbulent flow, such decay is not a property of the exact solution but reflects the error in the finite representation of

Table 1
Total mass in column i as a function of t.

i \ t	0	2	4	6	8
1	2	4.171	7.073	7.524	6.240
2	2	3.982	4.665	2.766	3.129
3	10	7.810	4.170	3.222	3.175
4	10	8.090	4.120	3.209	3.699
5	5	4.277	3.633	3.466	4.660
6	5	5.577	4.077	4.715	5.284
7	5	4.227	4.155	5.301	5.195
8	3	3.675	4.395	5.282	5.218
9	3	3.547	4.951	5.046	4.578
10	3	3.528	6.567	7.123	6.232

Table 2
Kinetic energy, potential energy, and total energy as functions of t, for a highly distorted initial surface.

t	KE	PE	E
0	0	155	155
0.5	9.56	152.5	162.0
1.0	17.63	148.1	165.7
1.5	23.84	140.6	164.5
2.0	30.60	132.3	162.9
2.5	36.20	126.2	162.4
3.0	36.05	122.4	158.4
3.5	33.92	121.0	154.9
4.0	29.19	121.7	150.8
4.5	23.83	124.6	148.4
5.0	24.49	127.4	151.9
5.5	20.95	128.8	149.8
6.0	18.85	128.1	147.0
6.5	17.30	126.8	144.1
7.0	19.32	125.0	144.4
7.5	20.37	123.8	144.1
8.0	22.67	123.0	145.7
8.5	21.25	122.7	143.9
9.0	18.50	122.1	140.6
9.5	16.67	121.5	138.2
10.0	15.19	121.3	136.5
10.5	13.99	121.1	135.1
11.0	13.49	120.5	134.0
11.5	13.55	119.0	132.5
12.0	15.10	116.5	131.6
12.5	15.19	114.7	129.9
13.0	14.34	113.8	128.2
13.5	12.45	113.7	126.2
14.0	11.62	114.3	125.9

the evolutionary equations. In this problem, because of the absence of any observed breaking or development of other pathologies in the flow, we are inclined to discount the presence of turbulence. The energy decay is such that we would tend not to give much credence to the quantitative computer results after time = 8, even apart from the unpersuasive character of the data at the side walls for earlier times.

COMPARISON WITH LINEAR THEORY

The algorithm presented in this report is decidedly inefficient when it comes to solving linear wave problems. First, in the linear regime the problem ceases to be a free-boundary problem in any important respects, and methods based on Green's function for the unperturbed domain are more effective. Also, a special burden is placed on the size of the numerical mesh, as it must be fine enough to resolve the linear displacements of the free surface, and yet the assumption of linearity requires that this be only a small portion of the vertical extent of the computational domain. Nevertheless, it is incumbent on us to compare the results of a calculation based on our algorithm with a known solution, and in this respect a linear problem naturally comes to mind.

The linear solution we compare with is the wave whose surface height is given by

$$z(x,t) = 5 + \cos \omega t \cos \frac{\pi x}{20}, \quad 0 \leq x \leq 20, \; t \geq 0, \qquad (104a)$$

where

$$\omega^2 = gk \tanh kh, \; g = 4, \; h = 5, \; k = \frac{\pi}{20}. \qquad (104b)$$

Our computation took place on the mesh

$$\Delta x_i = 1, \; 1 \leq i \leq 20, \qquad (105a)$$

$$\Delta z_j = \begin{cases} 1 & 1 \leq j \leq 3 \\ 0.5 & 4 \leq j \leq 11 \; . \\ 1 & 12 \leq j \leq 14 \end{cases} \qquad (105b)$$

Initial values of m_{ij} corresponding to the initial profile

$$z(x,0) = 5 + \cos \frac{\pi x}{20} , \tag{106a}$$

were provided, and the initial values of the momenta were

$$\mu_{xij} = \mu_{zij} = 0, \; 1 \le i \le 20, \; 1 \le j \le 14. \tag{106b}$$

These momenta were consistent with the fact that the fluid whose free surface is given by Eq. 104a is at rest at $t = 0$. Other relevant constants for the calculation were $\varepsilon = 10^{-5}$, $\varepsilon_1 = 0.005$, $\varepsilon_2 = 10^{-3}$, $\gamma_0 = 10$, $\rho_0 = 1$, $\tau = 0.1$, and $\Delta\alpha = 0.1$. The program was run up to time $t = 5.9$.

As we observed in our discussion of the first example, the z-coordinate of the free surface can be identified with the total mass in a column of fluid. Table 3 shows values of the total mass in each column for times $t = 0$, 0.5, 1.0, 1.5, 2.0, and 2.5. These are to be compared with the values (Eq. 104) predicted by linear theory and given in Table 4. As in the first example, we appear to get an accumulation of fluid at the side walls. This accumulation becomes noticeable after $t = 1.5$. By $t = 1.0$, we appear to be departing from the monotone dependence on x predicted by linear theory. This departure seems to commence at the side walls. The data for times $t > 2.5$ indicate greater departures from the linear solution. It is possible that genuine nonlinear effects should arise, since the initial height varies from 6 to 4 as x varies from 0 to 20. If the program we have described were considered to be a final product, we would be well advised to consider this point.

Table 5 gives the kinetic energy, potential energy, and total energy of the fluid as a function of time. A peculiar feature is that initially the potential energy varies only slightly, undergoing a slow steady decay. Of course, this behavior is not consistent with the linear theory. It is only after about $t = 4.4$ that the total energy remains essentially constant. The kinetic energy oscillates from one time to the next and also undergoes a slower oscillation, which has a peak with center around $t = 2.5$ and a low with center around $t = 4.8$. Analysis of the mass totals

Table 3
Total mass in column i as a function of t.

t i	0	0.5	1.0	1.5	2.0	2.5
1	5.997	5.956	5.875	5.759	5.614	5.406
2	5.972	5.913	5.748	5.492	5.184	4.889
3	5.924	5.861	5.715	5.552	5.338	5.123
4	5.853	5.795	5.651	5.437	5.191	4.927
5	5.760	5.708	5.581	5.401	5.197	5.021
6	5.649	5.609	5.504	5.328	5.135	4.930
7	5.523	5.496	5.402	5.263	5.108	4.948
8	5.383	5.356	5.290	5.186	5.057	4.909
9	5.233	5.208	5.158	5.091	5.013	4.911
10	5.078	5.073	5.056	5.032	4.986	4.914
11	4.922	4.932	4.951	4.958	4.945	4.909
12	4.767	4.774	4.811	4.861	4.899	4.905
13	4.617	4.627	4.680	4.777	4.873	4.909
14	4.478	4.515	4.556	4.647	4.766	4.886
15	4.351	4.396	4.514	4.592	4.692	4.816
16	4.240	4.274	4.424	4.564	4.648	4.757
17	4.147	4.191	4.326	4.599	4.735	4.819
18	4.076	4.124	4.248	4.483	4.876	4.999
19	4.028	4.077	4.197	4.379	4.688	5.217
20	4.003	4.062	4.207	4.442	4.847	5.554

Table 4

Position of the free surface for a linear wave at the center of cell i as a function of t.

i \ t	0	0.5	1.0	1.5	2.0	2.5
1	5.997	5.946	5.798	5.569	5.282	4.966
2	5.972	5.923	5.779	5.555	5.275	4.967
3	5.924	5.877	5.740	5.528	5.262	4.969
4	5.853	5.809	5.683	5.487	5.241	4.971
5	5.760	5.722	5.609	5.434	5.215	4.974
6	5.649	5.616	5.520	5.371	5.184	4.978
7	5.523	5.496	5.418	5.298	5.148	4.982
8	5.383	5.363	5.307	5.291	5.108	4.987
9	5.233	5.222	5.187	5.133	5.066	4.992
10	5.078	5.074	5.063	5.045	5.022	4.997
11	4.922	4.926	4.937	4.955	4.978	5.003
12	4.767	4.778	4.813	4.867	4.934	5.008
13	4.617	4.637	4.693	4.781	4.892	5.013
14	4.478	4.504	4.582	4.702	4.852	5.018
15	4.351	4.384	4.480	4.629	4.816	5.022
16	4.240	4.278	4.391	4.566	4.785	5.026
17	4.147	4.191	4.317	4.513	4.759	5.029
18	4.076	4.123	4.260	4.472	4.738	5.031
19	4.028	4.077	4.221	4.445	4.725	5.033
20	4.003	4.054	4.202	4.431	4.718	5.034

Table 5

Kinetic energy, potential energy, and total energy as functions of t, for a slightly distorted initial surface.

t	KE	PE	E
0	0	1021	1021
0.1	13.78	1018	1032
0.2	31.08	1018	1049
0.3	28.35	1017	1045
0.4	44.76	1016	1061
0.5	31.58	1016	1047
0.6	43.67	1015	1058
0.7	37.55	1014	1051
0.8	45.65	1013	1058
0.9	42.59	1012	1055
1.0	46.61	1011	1058
1.1	47.68	1010	1058
1.2	49.37	1010	1059
1.3	51.94	1009	1061
1.4	52.13	1009	1061
1.5	54.05	1008	1062
1.6	54.23	1008	1062
1.7	55.65	1008	1063
1.8	56.39	1008	1064
1.9	56.97	1007	1064
2.0	57.87	1007	1065
2.1	58.27	1007	1065
2.2	59.22	1006	1065
2.3	59.46	1005	1065
2.4	60.61	1005	1065
2.5	60.97	1004	1065
2.6	60.79	1003	1064
2.7	61.86	1003	1065
2.8	58.82	1004	1063
2.9	54.60	1004	1058
3.0	59.82	1004	1064
3.1	52.56	1005	1058
3.2	54.82	1005	1060
3.3	53.16	1006	1059
3.4	51.54	1006	1057
3.5	52.88	1006	1059
3.6	49.89	1006	1056
3.7	50.70	1005	1056
3.8	48.24	1005	1054
3.9	46.46	1005	1052
4.0	45.52	1005	1051
4.1	43.50	1006	1049
4.2	41.41	1006	1047
4.3	42.77	1006	1049
4.4	40.20	1007	1047
4.5	41.37	1007	1048
4.6	40.27	1007	1047
4.7	40.26	1007	1047
4.8	39.95	1007	1047
4.9	40.55	1007	1047
5.0	40.01	1007	1047
5.1	41.24	1007	1048
5.2	40.50	1007	1047
5.3	41.38	1006	1048
5.4	41.10	1006	1047
5.5	41.71	1006	1047
5.6	41.74	1005	1047
5.7	42.73	1005	1047
5.8	43.53	1004	1047
5.9	44.71	1003	1048

for the different columns of fluid and different times indicates a surface that is relatively flat at t = 2.5. By way of comparison, the linear theory predicts a surface that is flat at $t = \frac{\pi}{2\omega} \cong 2.447$. At this time the kinetic energy would be a maximum, and it would then decrease to 0 at $t = \frac{\pi}{\omega}$. In our calculation, the "minimum" kinetic energy is about two-thirds the "maximum" value. Thus, from the point of view of location of the free surface and period of oscillation, our calculation gives results as good as can be expected for the grid we have used; but from the point of view of energy balance, the picture is not as satisfactory.

Another feature of the flow that can be compared with the prediction of linear theory is the "pressure." We do not compute a pressure, but in the interior of the region of flow the quantity $\frac{2}{\tau^2} v$ takes the place of the pressure for classical hydrodynamic flows (Ref. 1). In the linear theory, the pressure should be essentially the hydrostatic pressure, or $\rho_0 g$ times the distance below the free surface. In Table 6 we give $v_{11,1}$ and $\sum_{j=1}^{14} m_{11,j}$ as a function of time. After some initial oscillation, the values of v settle down around t = 1. Thereafter, they appear to agree very well with the values predicted by the linear theory.

(Note that Eqs. 12 and 10d and e imply that, for the analytical algorithm, $\nabla v \cdot n = 0$ for $x \in \partial D$. This is not true of the the pressure, and, in fact, $\frac{2}{\tau^2} v$ will differ from the pressure in a layer of thickness $\frac{\tau^2}{2} g$ near z = 0 where v_z will jump from 0 to $-\frac{\tau^2}{2} g \rho_0$.)

As we observe from Eq. 26, for the computational scheme described in this report, v will be a rough measure of the computational time taken in solving the Stefan problem (Eq. 10). This is the longest part of the calculation, and accordingly we may expect the computational time overall to increase with the size of the expected values of v, when the method of solution is the one we have used to date. In the next section, we will discuss more efficient ways of computing v.

Table 6

Total mass between x = 10 and x = 11 and $v_{11,1}$ as functions of t.

t	$\sum_{j=1}^{14} m_{11,j}$	$v_{11,1}$
0.1	4.919	0.12901
0.2	4.921	0.10557
0.3	4.923	0.07182
0.4	4.928	0.13092
0.5	4.932	0.05875
0.6	4.935	0.12553
0.7	4.940	0.07098
0.8	4.944	0.12000
0.9	4.948	0.07786
1.0	4.951	0.10537
1.1	4.953	0.09245
1.2	4.955	0.09562
1.3	4.957	0.09915
1.4	4.958	0.09356
1.5	4.958	0.09809
1.6	4.957	0.09404
1.7	4.956	0.09700
1.8	4.954	0.09573
1.9	4.950	0.09536
2.0	4.945	0.09534
2.1	4.940	0.09494
2.2	4.933	0.09601
2.3	4.926	0.09381
2.4	4.918	0.09579
2.5	4.909	0.09476
2.6	4.900	0.09416
2.7	4.890	0.10167
2.8	4.880	0.09101
2.9	4.871	0.08565
3.0	4.861	0.11353
3.1	4.850	0.07945
3.2	4.840	0.10154
3.3	4.829	0.09005
3.4	4.818	0.09396
3.5	4.807	0.09789
3.6	4.797	0.08945
3.7	4.789	0.10097
3.8	4.783	0.09340
3.9	4.779	0.09755
4.0	4.778	0.09741
4.1	4.780	0.09830
4.2	4.786	0.09385
4.3	4.795	0.10251
4.4	4.813	0.09087
4.5	4.841	0.10141
4.6	4.885	0.09525
4.7	4.940	0.09883
4.8	4.993	0.09677
4.9	5.031	0.10034
5.0	5.051	0.09633
5.1	5.064	0.10168
5.2	5.074	0.09318
5.3	5.083	0.09861
5.4	5.092	0.09601
5.5	5.103	0.09886
5.6	5.117	0.09541
5.7	5.133	0.09870
5.8	5.148	0.09665
5.9	5.164	0.09752

Since the linear flow is irrotational, a further test of the accuracy of our algorithm would be a check on the vorticity of the computed flow. The same sort of test might also be performed on the data of the first example, since we would expect that flow to be irrotational in the absence of breaking. We have not examined the vorticity for these flows in detail because the preliminary nature of our results does not seem to warrant it at this time.

COLLISION OF STREAMS WITH JET FORMATION

We performed three runs, with different computational meshes and time steps, for the flow corresponding to the initial conditions

$$p(x,z) = \begin{cases} \rho_0 & 0 < x < 2,\ 0 < z < 7 \\ 0 & 0 < x < 2,\ z > 7 \\ \rho_0 & 2 < x < 5,\ 0 < z < 2 \\ 0 & 2 < x < 5,\ z > 2 \end{cases} \qquad (107a)$$

$$(\rho u)(x,z) = \left| \begin{array}{ll} -10\rho(x,z) & 0 < z < 2 \\ 0 & z > 2 \end{array} \right. , \qquad (107b)$$

and

$$(\rho w)(x,z) = \begin{cases} -10\rho(x,z) & 0 < x < 2,\ z > 2 \\ 0 & 0 < x < 2,\ 0 < z < 2\ . \\ 0 & x > 2 \end{cases} \qquad (107c)$$

For all three runs, we had $g = 1$, $\rho_0 = 1$, $\varepsilon = 10^{-5}$, $\varepsilon_1 = 10^{-2}$, $\varepsilon_2 = 10^{-3}$, and $\gamma_0 = 10$. Otherwise, we had for run 1

$$\tau = 0.1,\ \Delta x_i = 1,\ \Delta z_j = 1,\ I = 5,\ J = 10,\ \Delta\alpha = 0.1, \tag{108a}$$

and we ran the problem 10 time steps; for run 2

$$\tau = 0.05,\ \Delta x_i = 0.5,\ \Delta z_j = 0.5,\ I = 10,\ J = 20,\ \Delta\alpha = 0.05, \tag{108b}$$

and the problem was run 20 time steps; for run 3

$$\tau = 0.025,\ \Delta x_i = 0.25,\ \Delta z_j = 0.25,\ I = 20,\ J = 40,\ \Delta\alpha = 0.025, \tag{108c}$$

and we computed the flow for 40 time steps.

Tables 7, 8, and 9 record the kinetic energy, potential energy, and total energy as functions of time for the three runs. Generally we observe that the energy tends to be higher at a given time for the run with the finer computational mesh. This is in accordance with our observation at the beginning of this section that the finite grid leads to a spurious energy loss through diffusion and collisions. However, beyond this energy loss there appears to be an energy loss for $t < 0.3$ *that is not related to the finite grid* spacing but that may reflect the presence of turbulence in the flow. After about $t = 0.3$ the slower diminution of energy observed may be due primarily to the error inherent in the discretization. (However, the exact solution would still be expected to exhibit energy loss associated with the collapse of cavities after $t = 0.3$.) As we observed in the second example, in all three runs the potential energy varies slowly.

The three runs were compared for their consistency in depicting the free surface at a given time. The hope is that one can get a measure of the error in a computation by examining the dependence of the output on the mesh size. Of course, agreement of calculations and the demonstration of their convergence says nothing about what they converge to. That is a task for the theory. We chose to be rather crude in plotting the free surfaces obtained in order not to give the numerical results any particular advantage. Our criterion for drawing a free surface is as follows: If ρ_{ij} as computed in Eq. 39 is $\geq \frac{1}{2}$, the cell is included in the "water" region; if $\rho_{ij} < \frac{1}{2}$, the cell is in the "vacuum" region.

Table 7

Kinetic energy as a function of t for three runs.

t	Run 1	Run 2	Run 3
0	1000	1000	1000
0.025			628.12
0.050		483.74	575.56
0.075			519.95
0.100	361.23	404.73	481.68
0.125			431.86
0.150		349.13	392.15
0.175			326.13
0.200	274.86	254.41	301.95
0.225			262.88
0.250		218.99	238.81
0.275			221.39
0.300	129.25	170.10	209.67
0.325			200.55
0.350		144.44	193.09
0.375			186.75
0.400	92.54	130.18	180.31
0.425			174.73
0.450		122.11	171.17
0.475			168.95
0.500	74.51	114.61	167.16
0.525			164.04
0.550		109.11	158.33
0.575			149.79
0.600	65.99	103.28	138.34
0.625			130.86
0.650		96.00	116.01
0.675			107.38
0.700	60.91	87.38	101.45
0.725			95.54
0.750		76.43	87.07
0.775			73.58
0.800	52.89	59.65	64.65
0.825			59.69
0.850		53.32	55.29
0.875			51.83
0.900	44.53	48.28	47.85
0.925			44.25
0.950		42.93	43.40
0.975			42.29
1.000	39.36	39.81	42.47

Table 8
Potential energy as a function of t for three runs.

t	Run 1	Run 2	Run 3
0	55	55	55
0.025			53.70
0.050		52.78	53.51
0.075			53.38
0.100	51.36	52.70	53.25
0.125			53.11
0.150		52.66	52.97
0.175			52.84
0.200	51.89	52.59	52.69
0.225			52.52
0.250		52.48	52.34
0.275			52.18
0.300	52.48	52.32	52.01
0.325			51.82
0.350		52.13	51.64
0.375			51.45
0.400	53.05	51.89	51.21
0.425			50.96
0.450		51.59	50.74
0.475			50.51
0.500	53.51	51.20	50.24
0.525			49.83
0.550		50.73	49.18
0.575			48.33
0.600	53.80	50.06	47.39
0.625			46.66
0.650		49.25	46.19
0.675			45.80
0.700	53.95	48.68	45.46
0.725			45.12
0.750		48.82	44.75
0.775			44.29
0.800	52.80	47.81	44.21
0.825			44.05
0.850		47.53	43.84
0.875			43.66
0.900	52.12	47.37	43.40
0.925			43.12
0.950		47.17	43.30
0.975			43.40
1.000	51.62	46.88	43.52

Table 9

Total energy as a function of t for three runs.

t	Run 1	Run 2	Run 3
0	1055	1055	1055
0.025			681.82
0.050		536.52	629.07
0.075			519.95
0.100	412.59	457.43	534.93
0.125			484.97
0.150		401.79	445.12
0.175			378.97
0.200	326.75	307.00	354.64
0.225			315.40
0.250		271.46	291.16
0.275			273.57
0.300	181.72	222.42	261.68
0.325			252.37
0.350		196.56	244.72
0.375			238.20
0.400	145.54	182.07	231.52
0.425			225.69
0.450		173.70	221.92
0.475			219.46
0.500	128.02	165.81	217.40
0.525			213.86
0.550		159.84	207.51
0.575			198.13
0.600	119.79	153.34	185.73
0.625			177.52
0.650		145.25	162.20
0.675			153.19
0.700	114.86	136.07	146.91
0.725			140.66
0.750		125.25	131.81
0.775			117.87
0.800	105.69	107.46	108.85
0.825			103.75
0.850		100.85	99.12
0.875			95.50
0.900	96.65	95.65	91.25
0.925			87.37
0.950		90.10	86.71
0.975			85.69
1.000	90.98	86.69	85.99

Figure 1 shows the initial water surface. Figures 2 through 31 depict computed water surfaces for various times and various runs. In general, the agreement among the figures, especially those for runs 2 and 3, is rather good. Although we cannot have too much faith in the runs for later times because of the loss of energy, it is still not unreasonable to expect them to exhibit correctly some of the qualitative features of the flow. Thus, we may expect that for the actual flow a cavity appears in the left interior around $t = 0.1$, and that by $t = 0.2$ a jet has struck the right wall $x = 5$ and a cavity has been formed there. The interior cavity at the left disappears between $t = 0.5$ and $t = 0.8$, and the cavity at the right closes in at the wall around $t = 0.7$. By $t = 0.8$ the larger cavities have closed in. One's intuition might lead one to expect the jet to bounce off the right wall to create a leftward moving jet. Indeed, we see a hint of such behavior in Figs. 10 and 13. It is possible that the program has suppressed this tendency by allowing fluid to accumulate at the right-hand wall instead, in a manner reminiscent of the computed flow for the first two examples above.

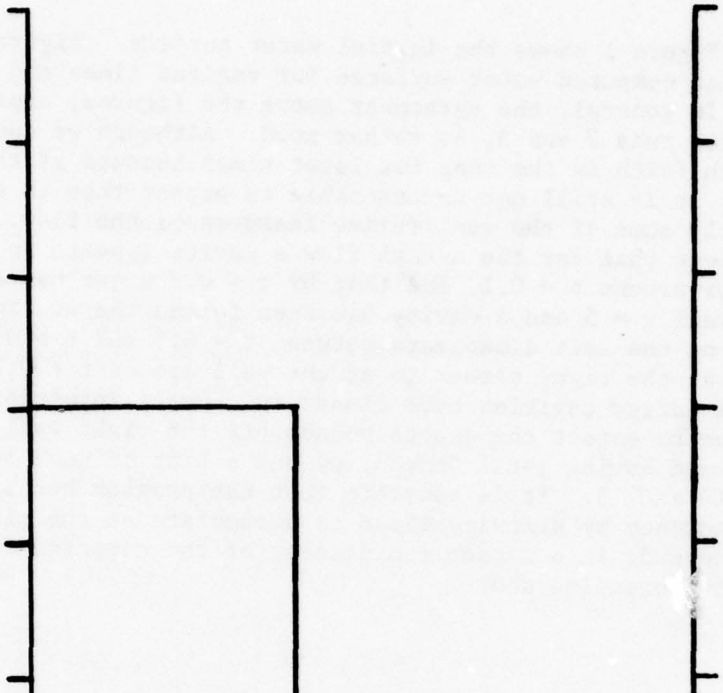


Fig. 1 Initial water surface at time t = 0.

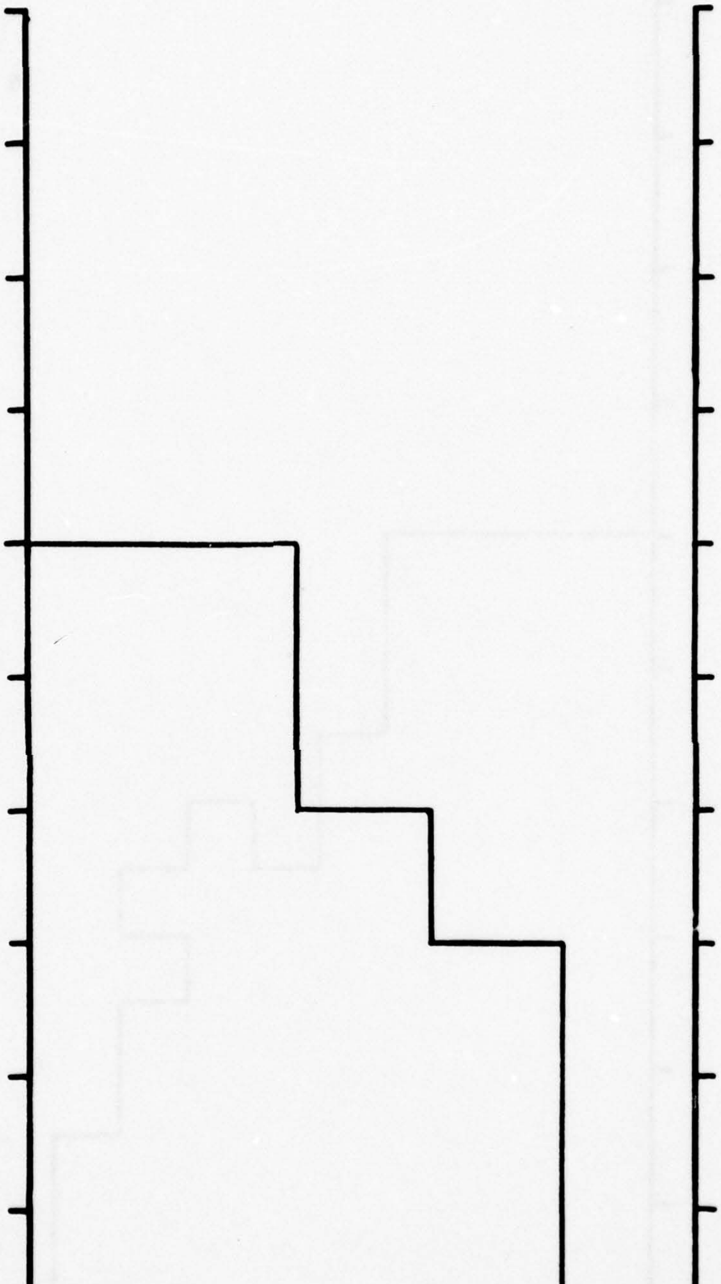


Fig. 2 Water surface for run 1 at time t = 0.1.

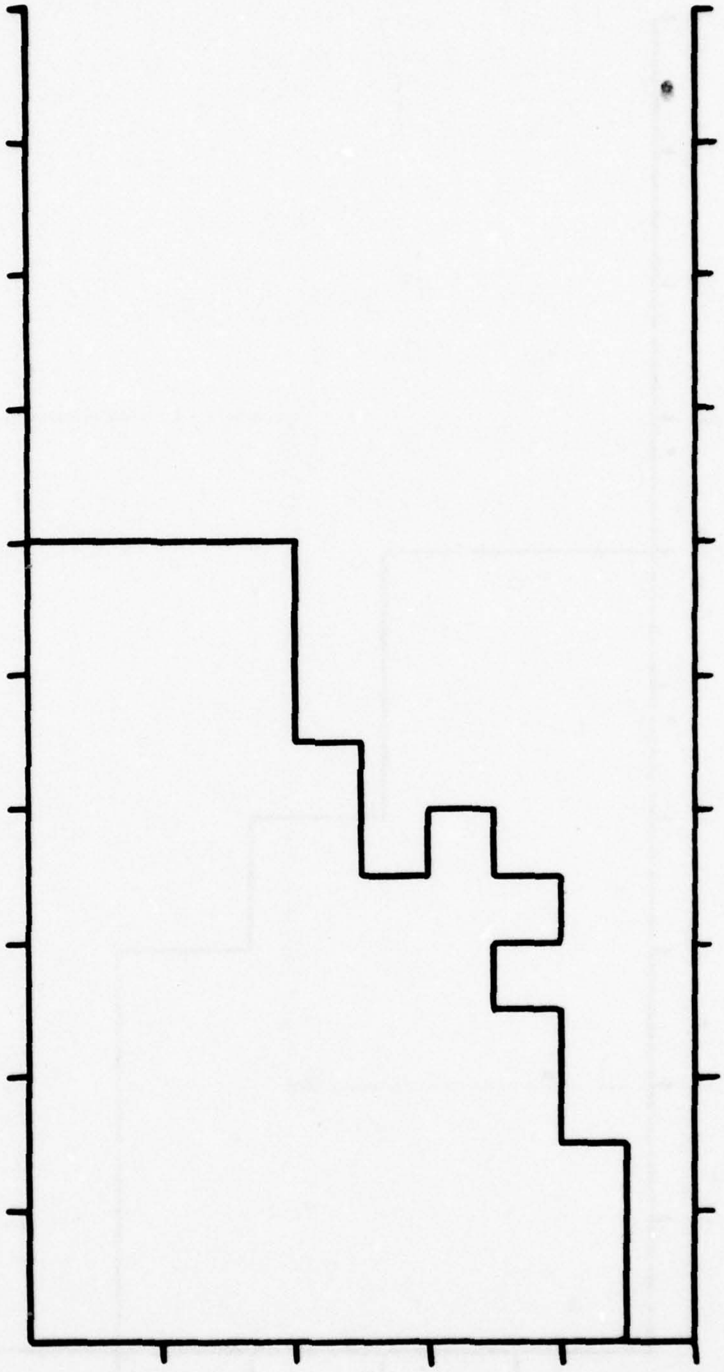


Fig. 3 Water surface for run 2 at time t = 0.1.

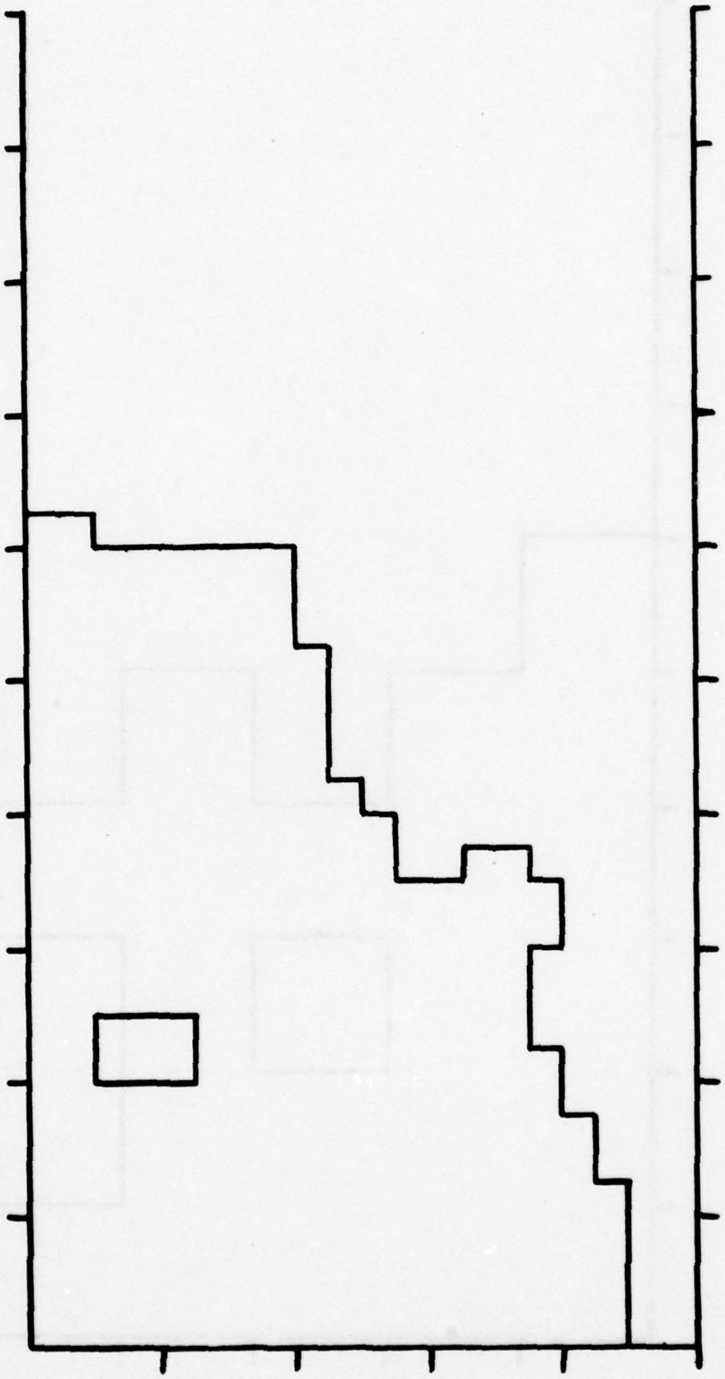


Fig. 4 Water surface for run 3 at time t = 0.1.

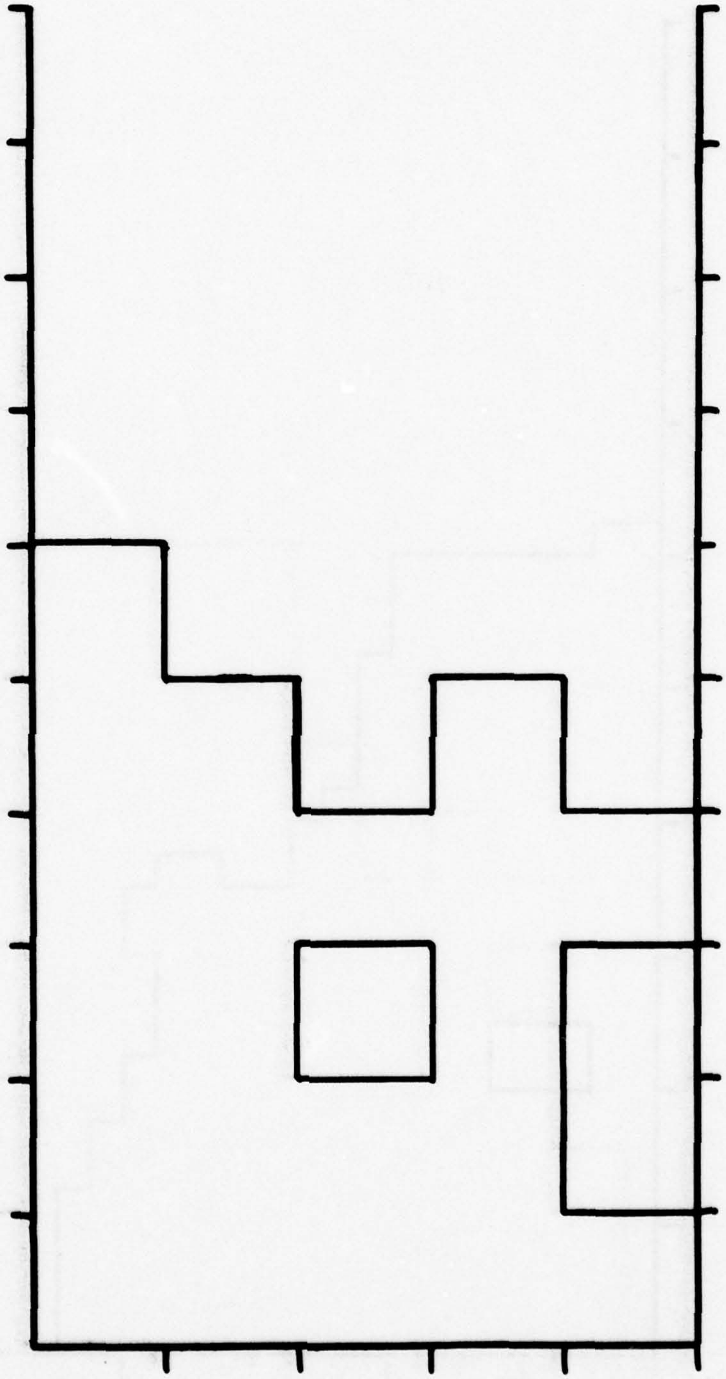


Fig. 5 Water surface for run 1 at time t = 0.2.

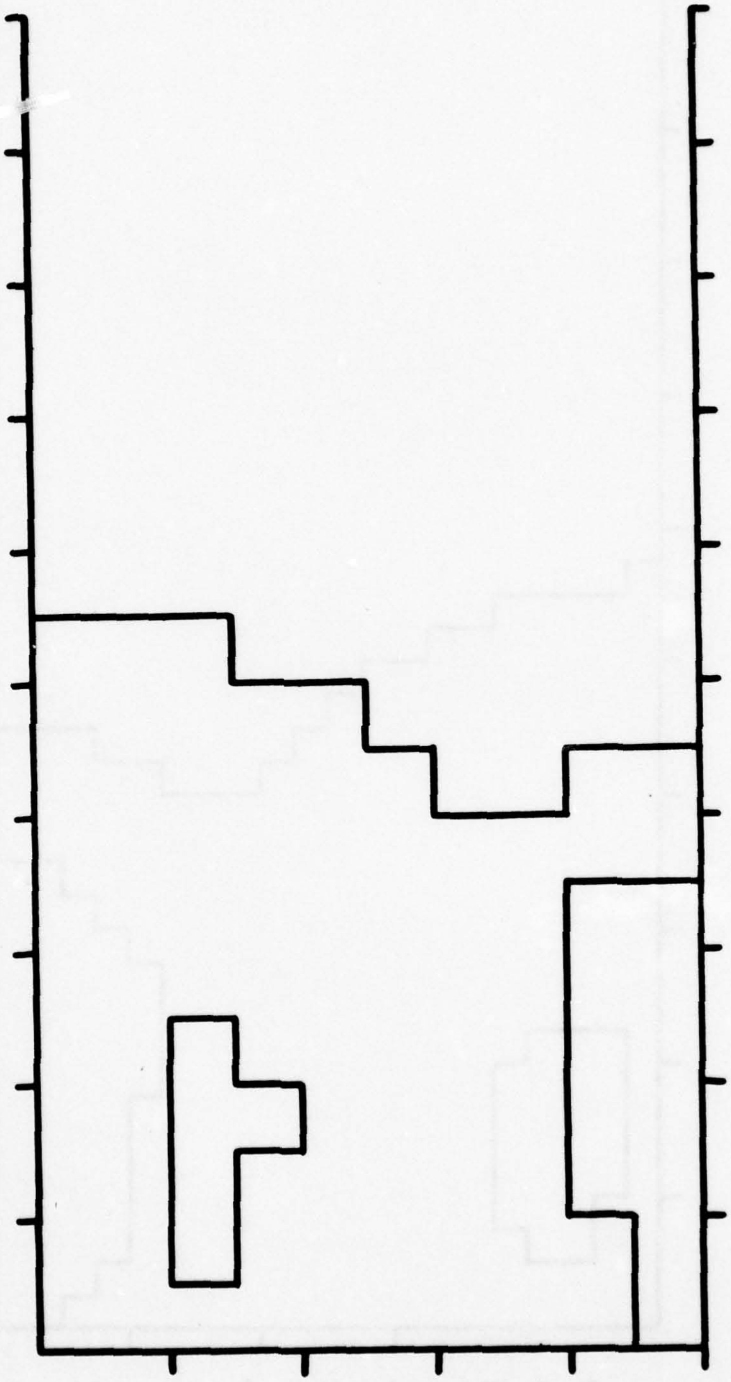


Fig. 6 Water surface for run 2 at time t = 0.2.

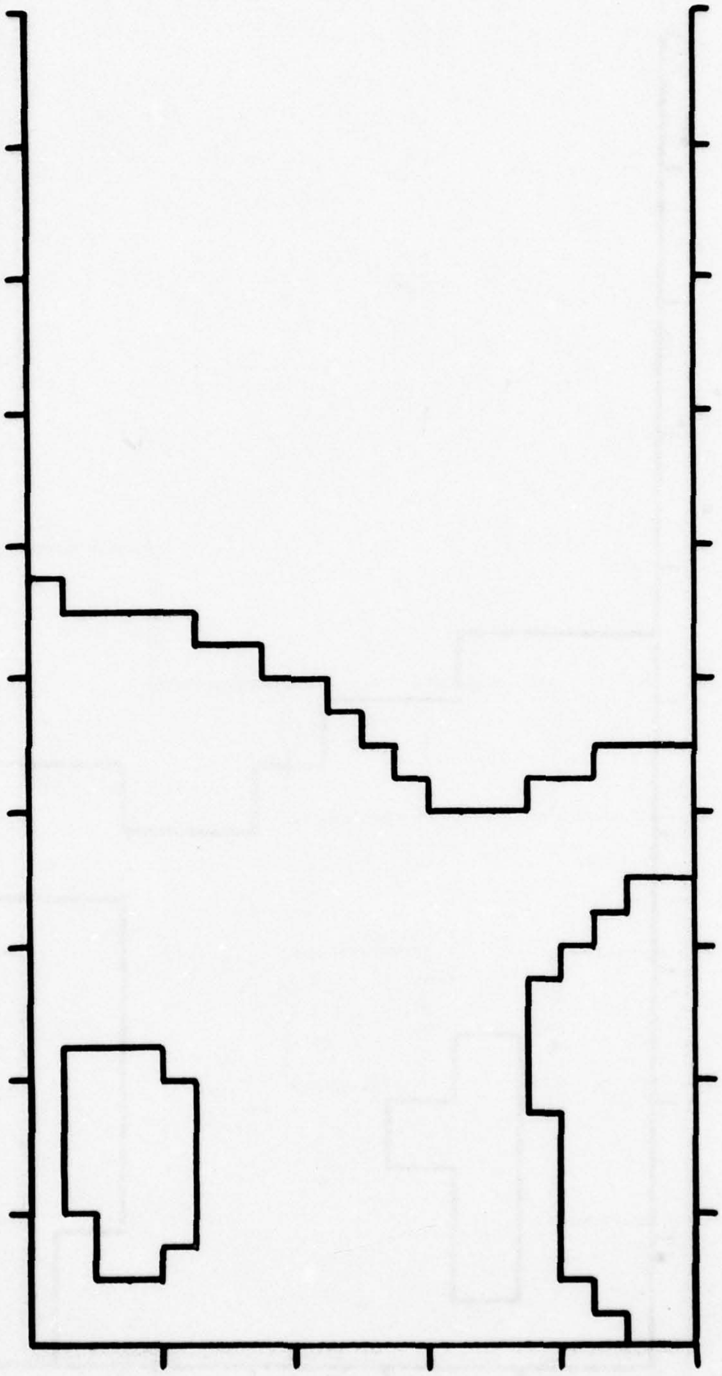


Fig. 7 Water surface for run 3 at time t = 0.2.

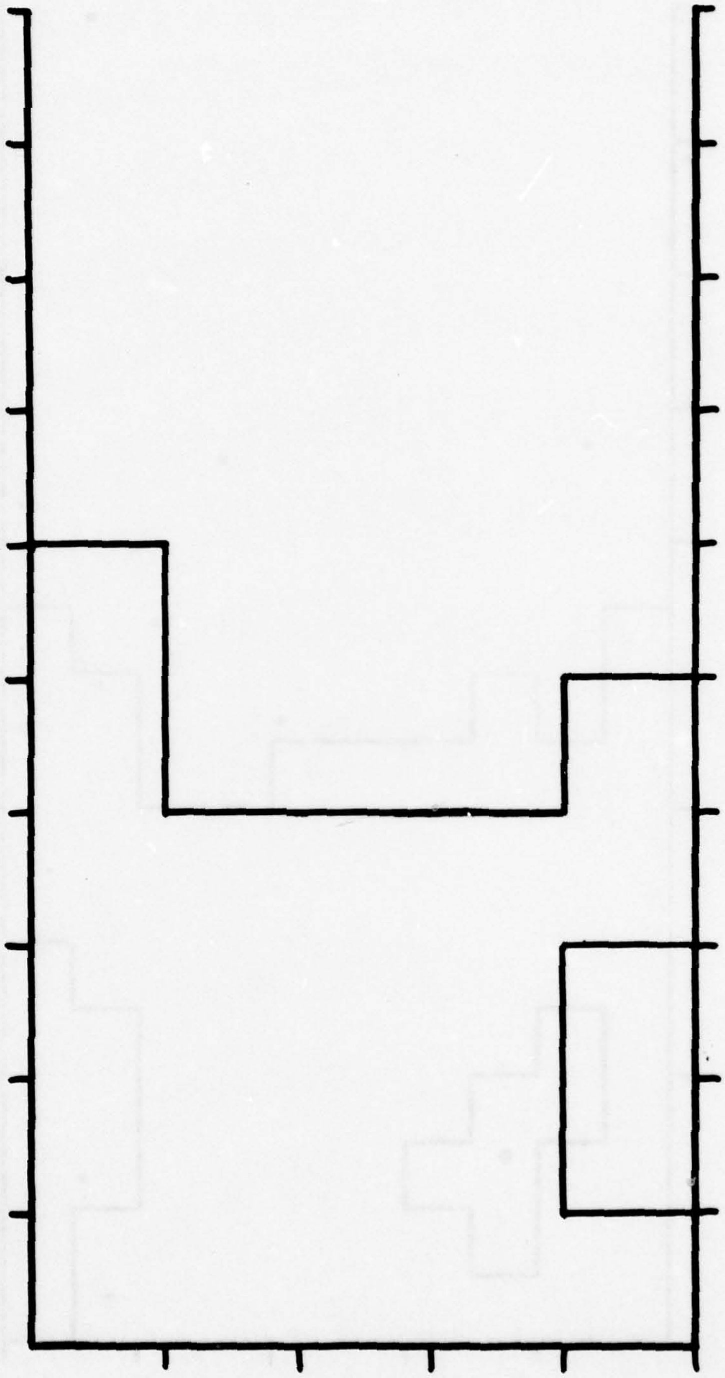


Fig. 8 Water surface for run 1 at time t = 0.3.

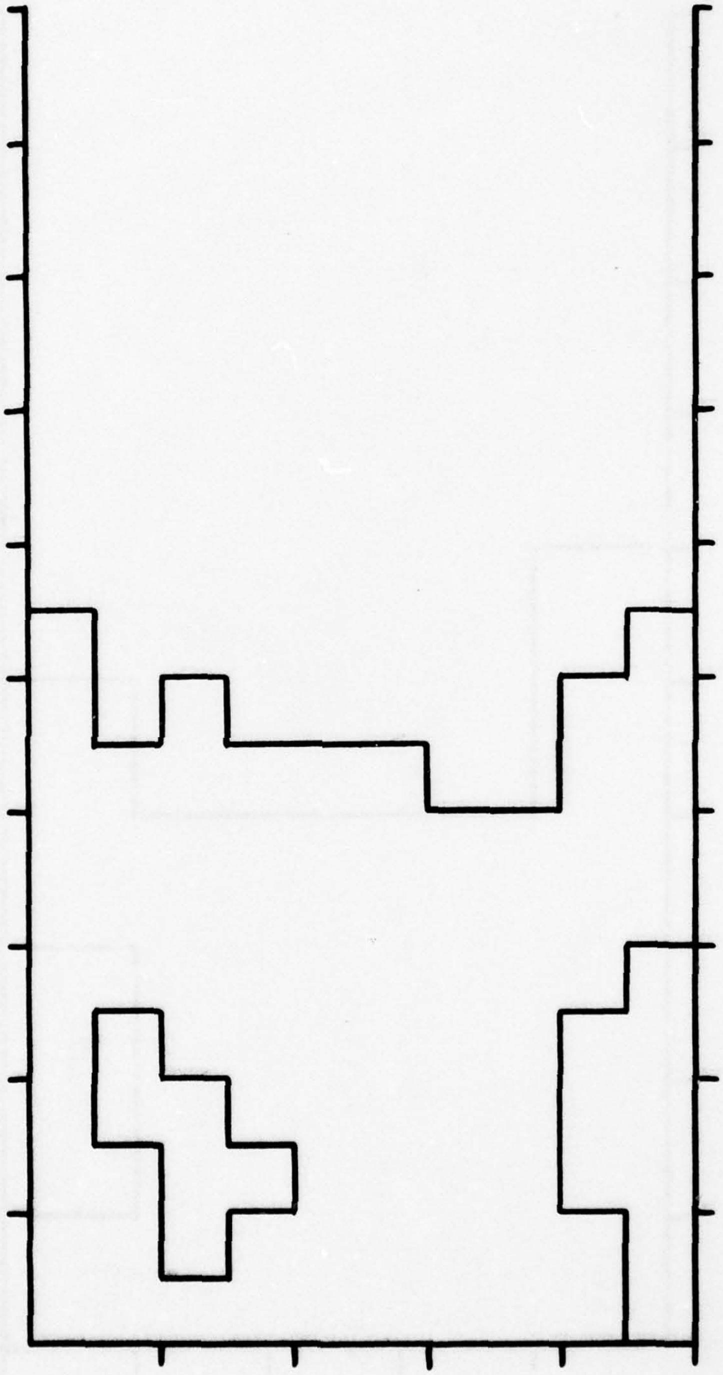


Fig. 9 Water surface for run 2 at time t = 0.3.

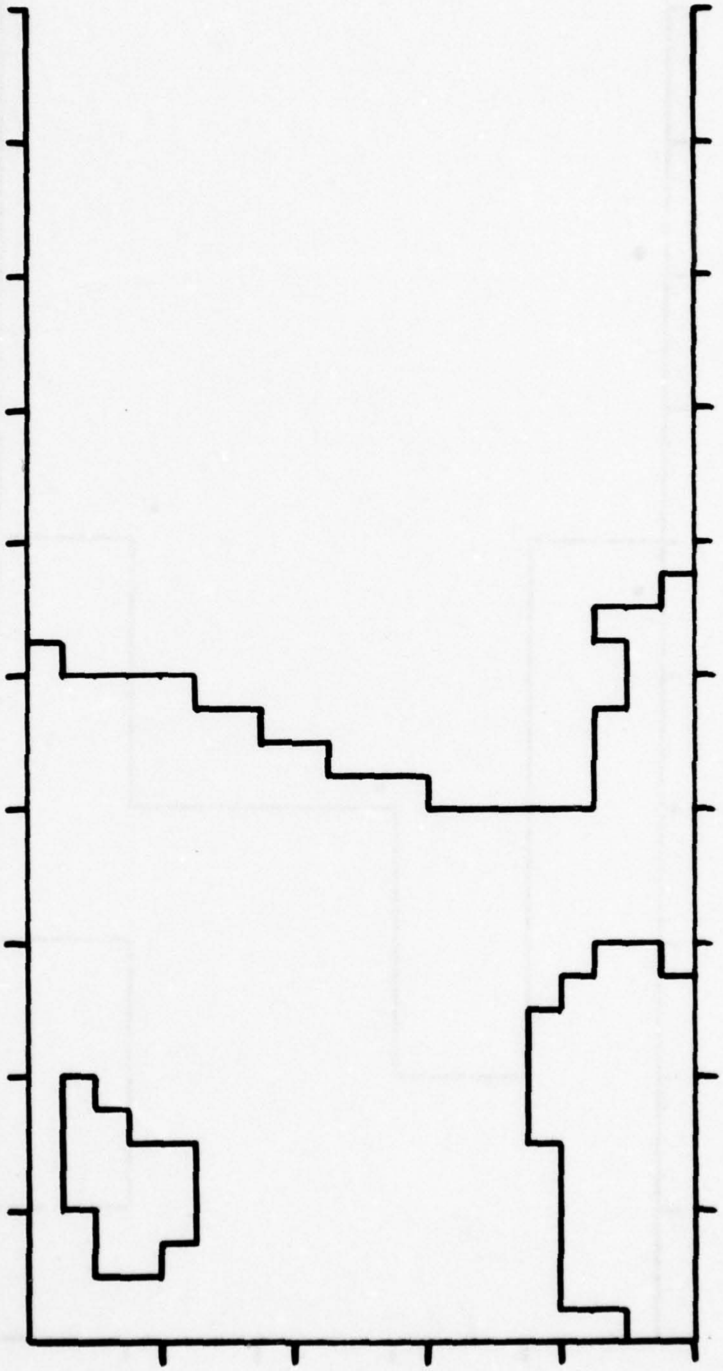


Fig. 10 Water surface for run 3 at time t = 0.3.

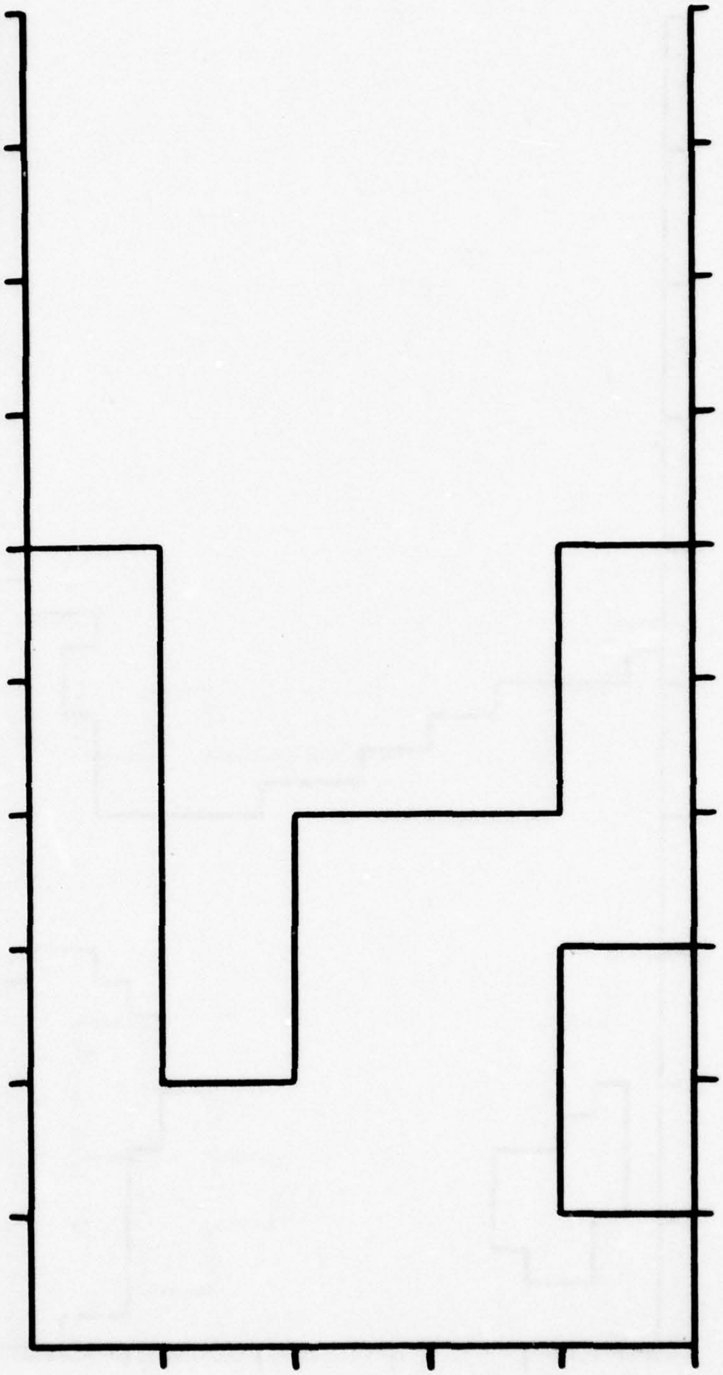


Fig. 11 Water surface for run 1 at time t = 0.4.

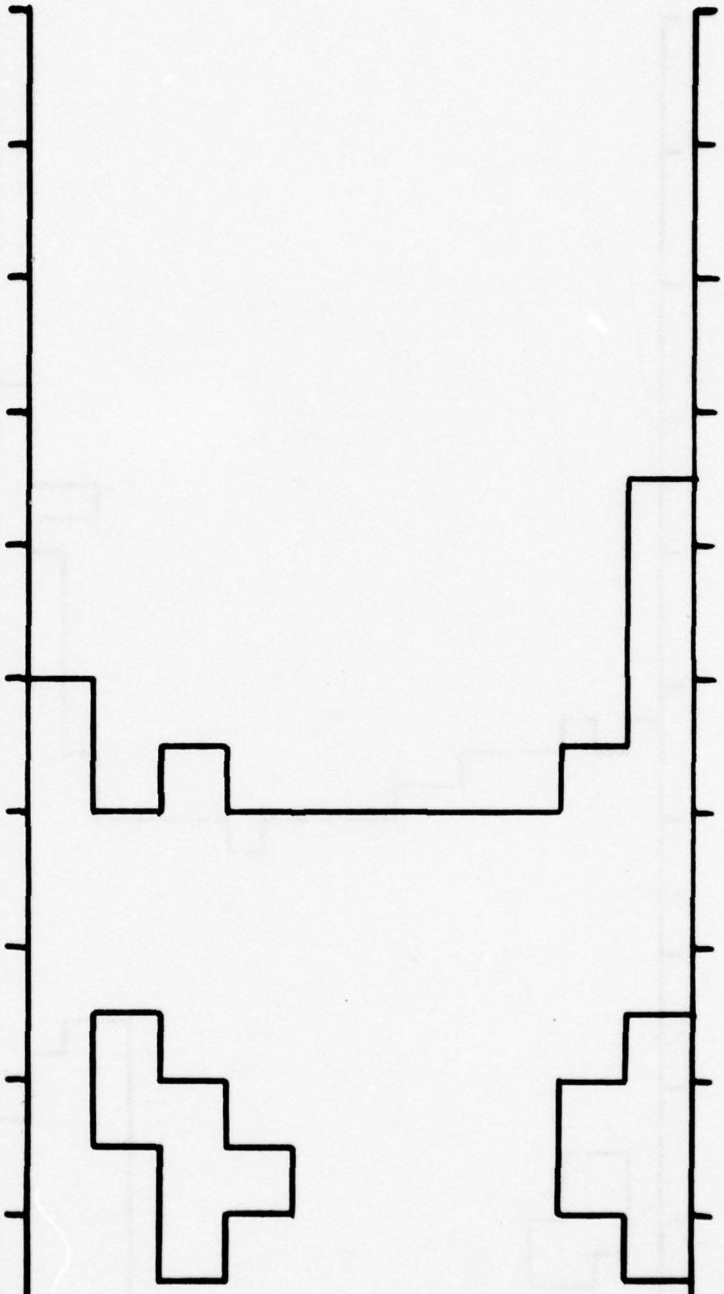


Fig. 12 Water surface for run 2 at time t = 0.4.

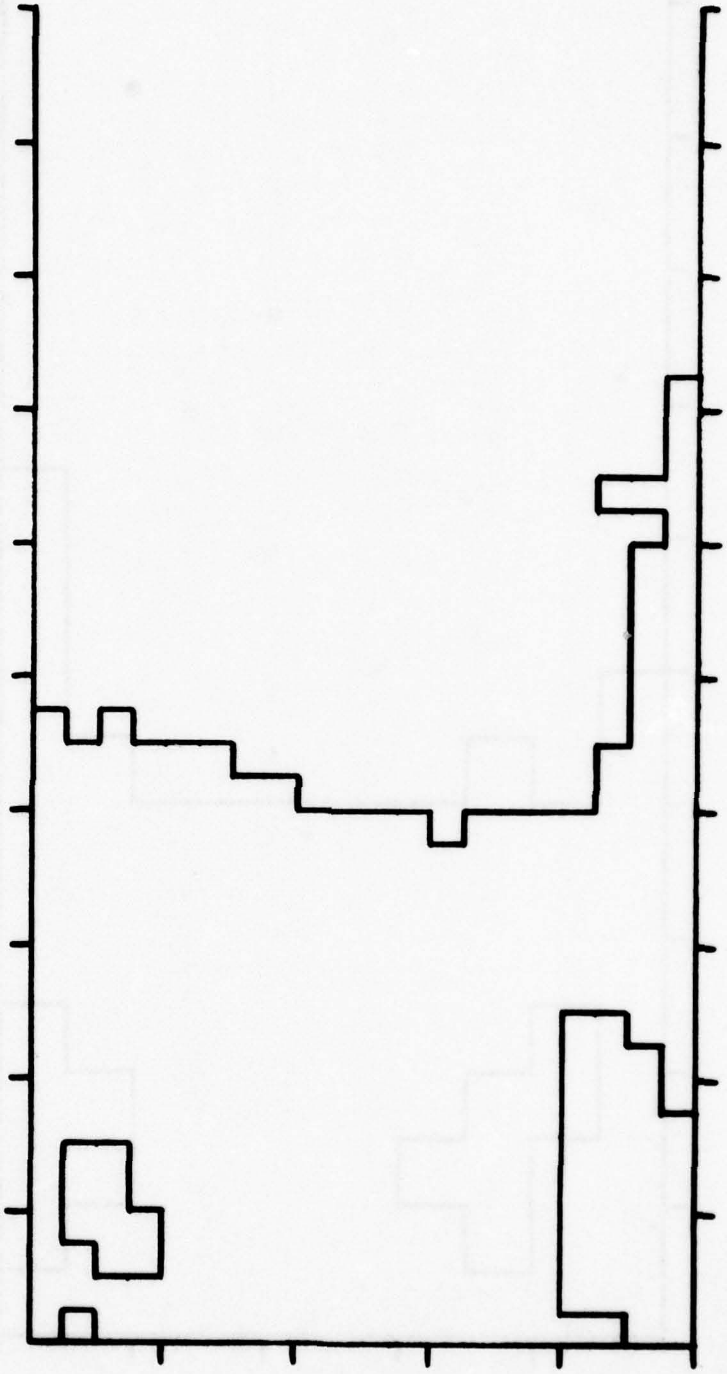


Fig. 13 Water surface for run 3 at time t = 0.4.

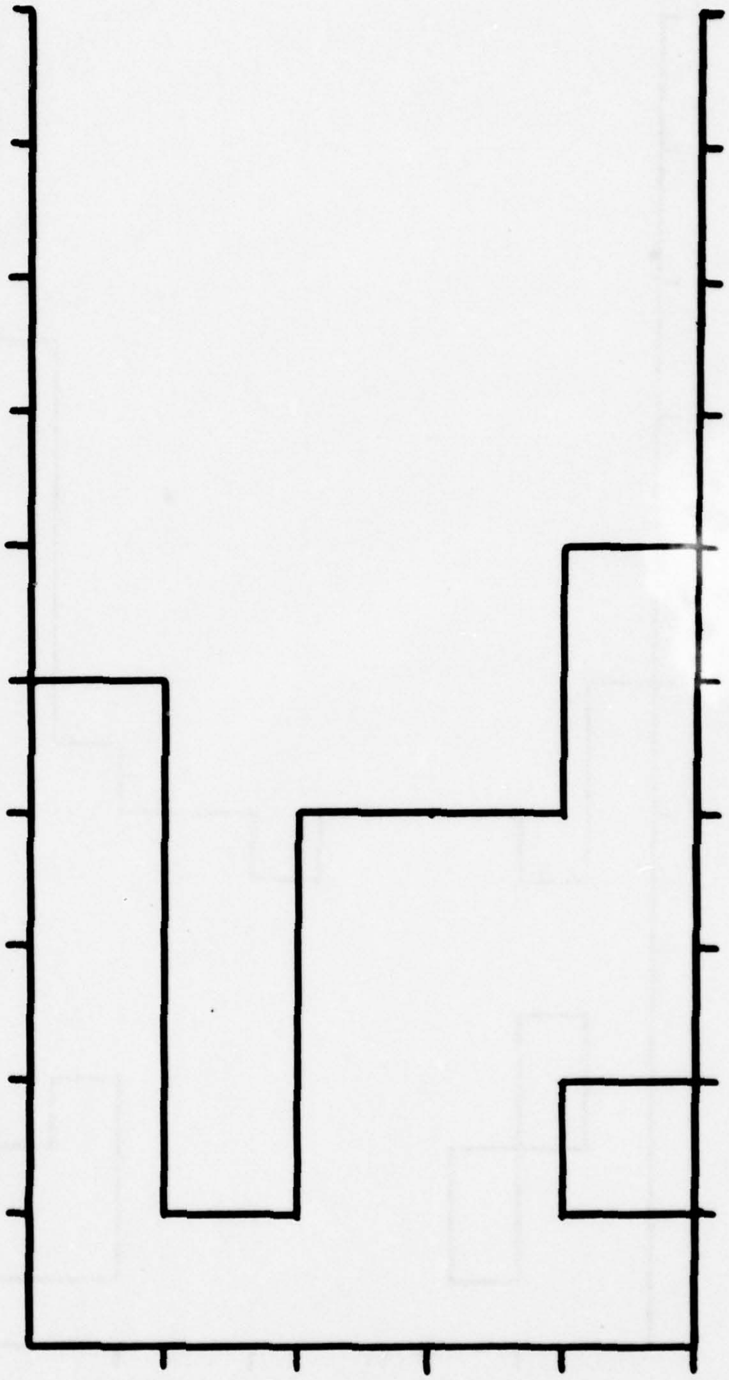


Fig. 14 Water surface for run 1 at time t = 0.5.

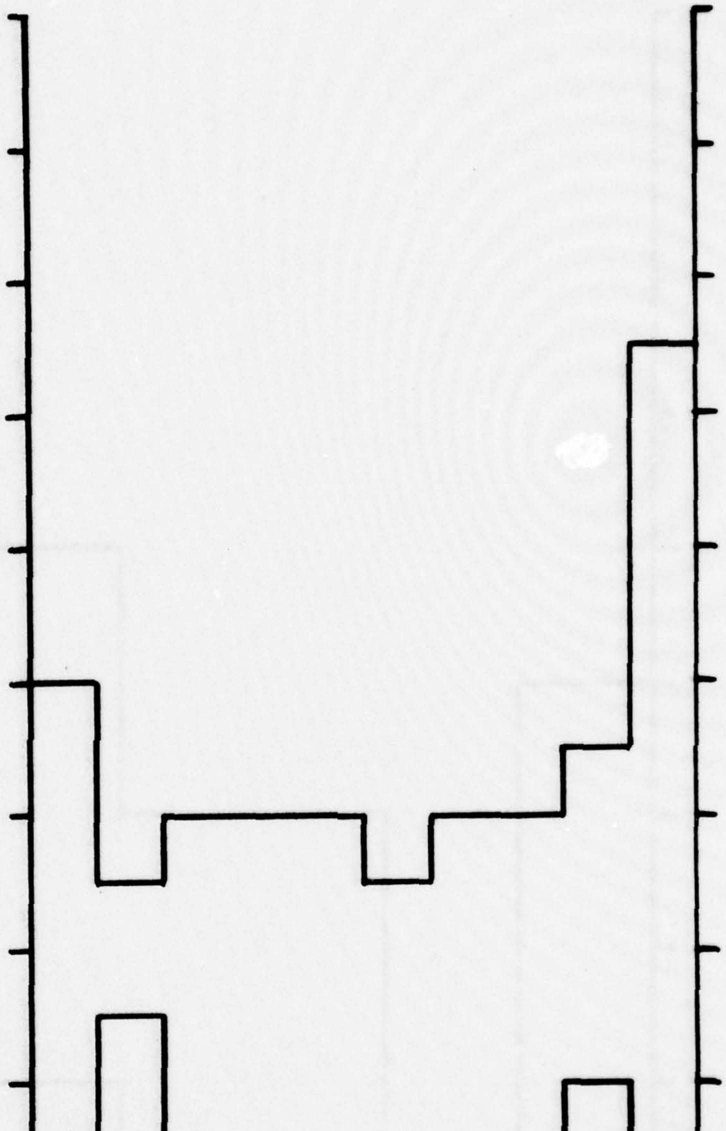


Fig. 15 Water surface for run 2 at time t = 0.5.

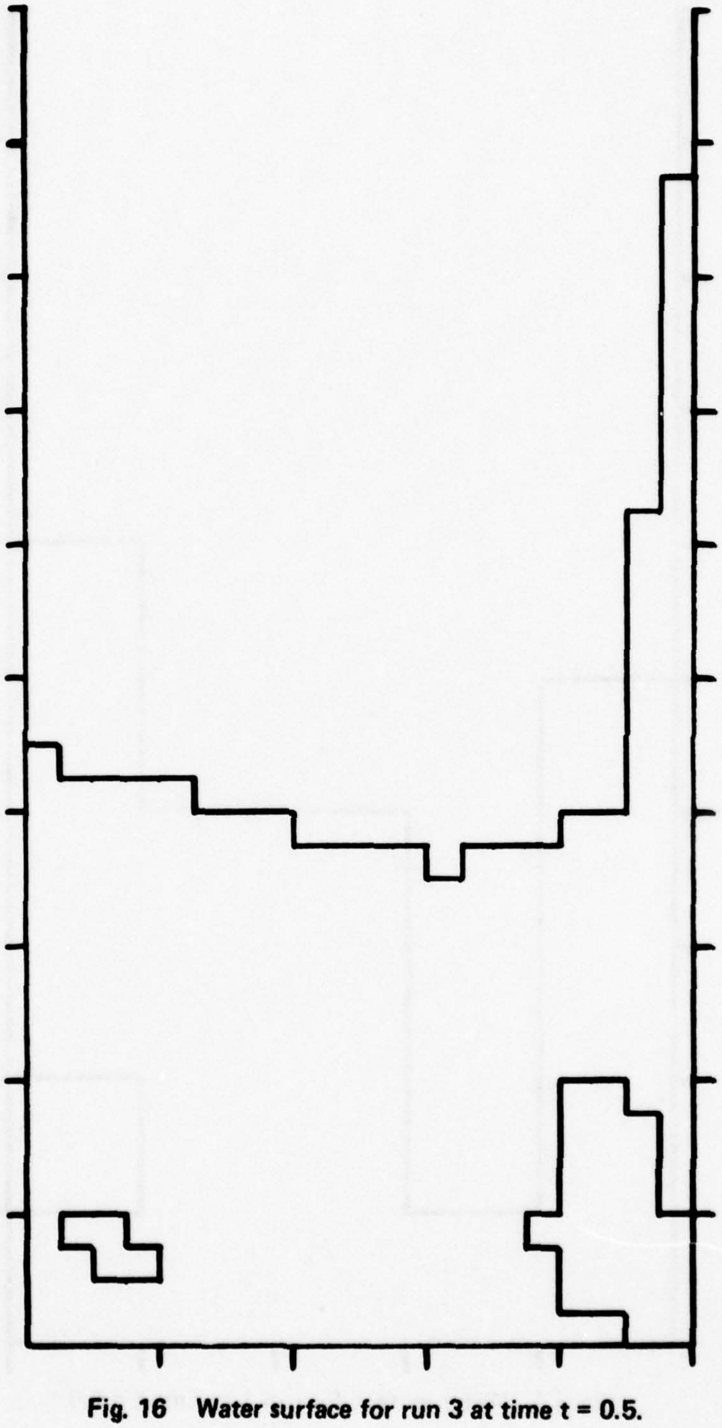


Fig. 16 Water surface for run 3 at time t = 0.5.

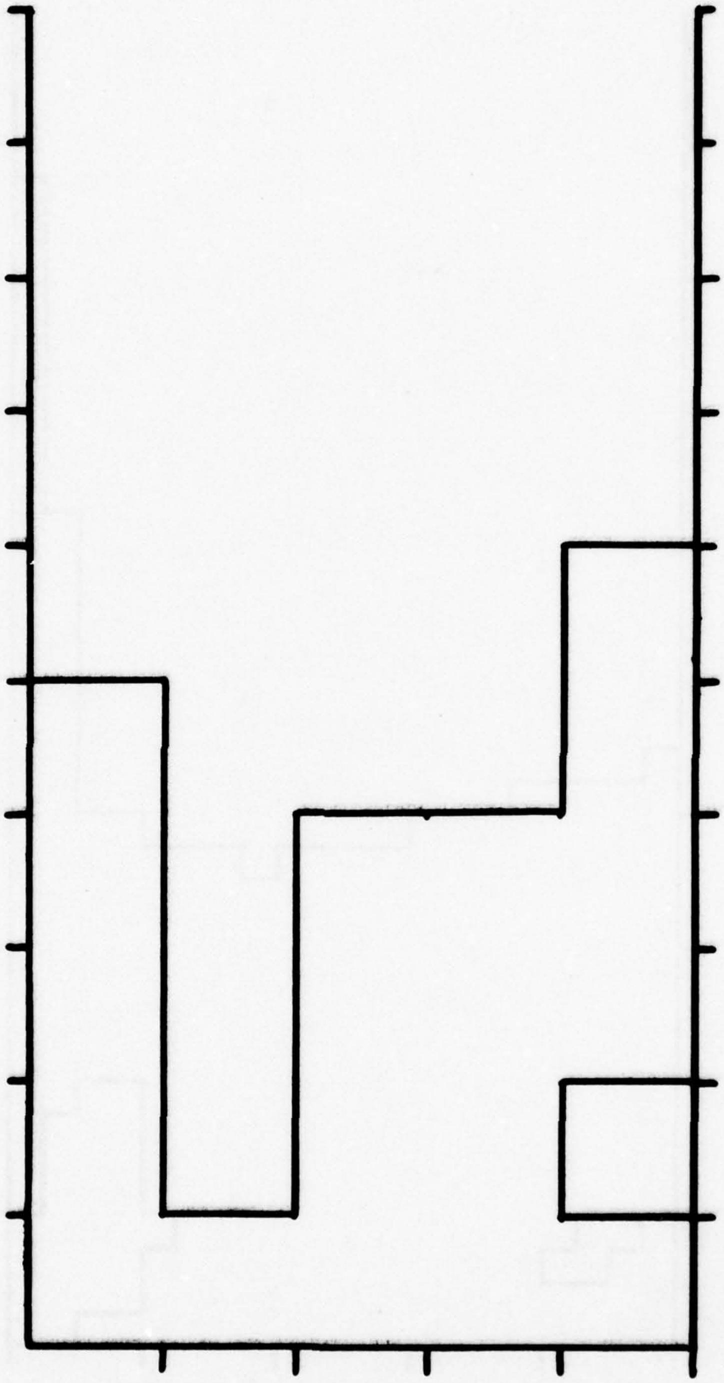


Fig. 17 Water surface for run 1 at time t = 0.6.

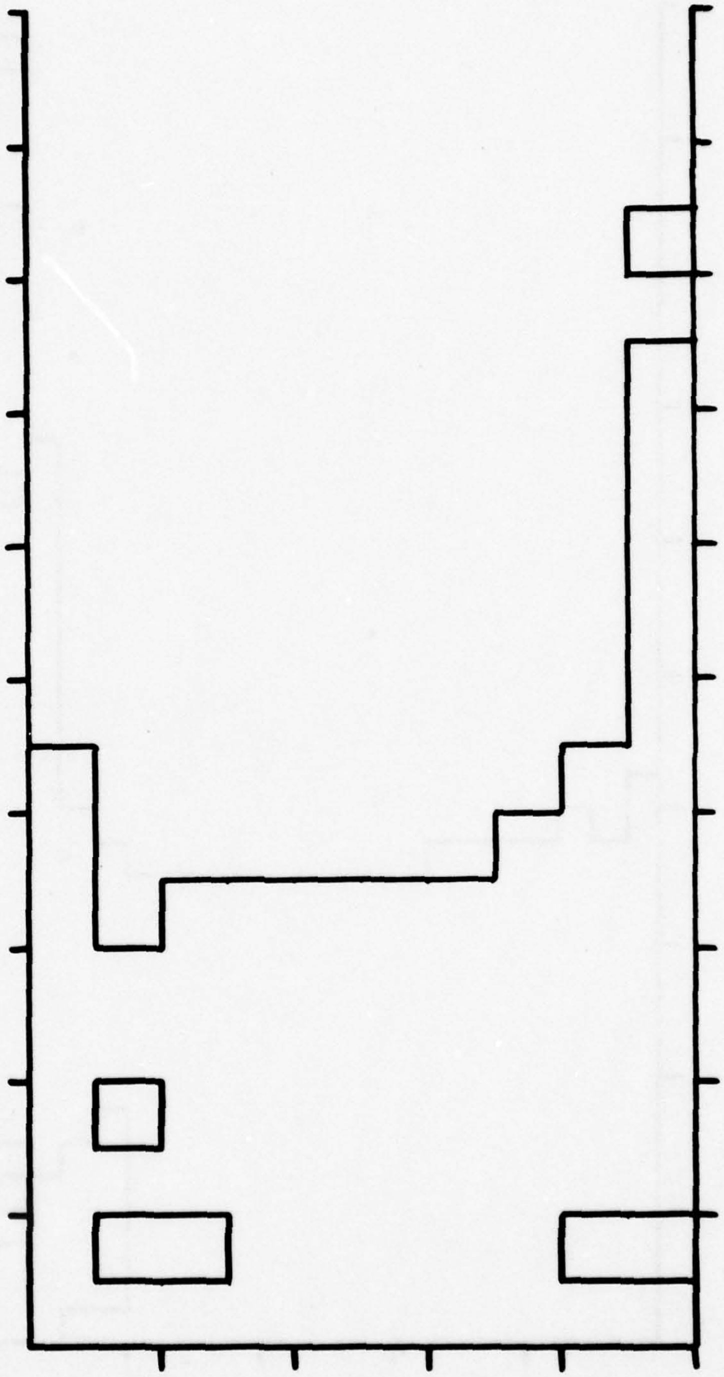


Fig. 18 Water surface for run 2 at time t = 0.6.

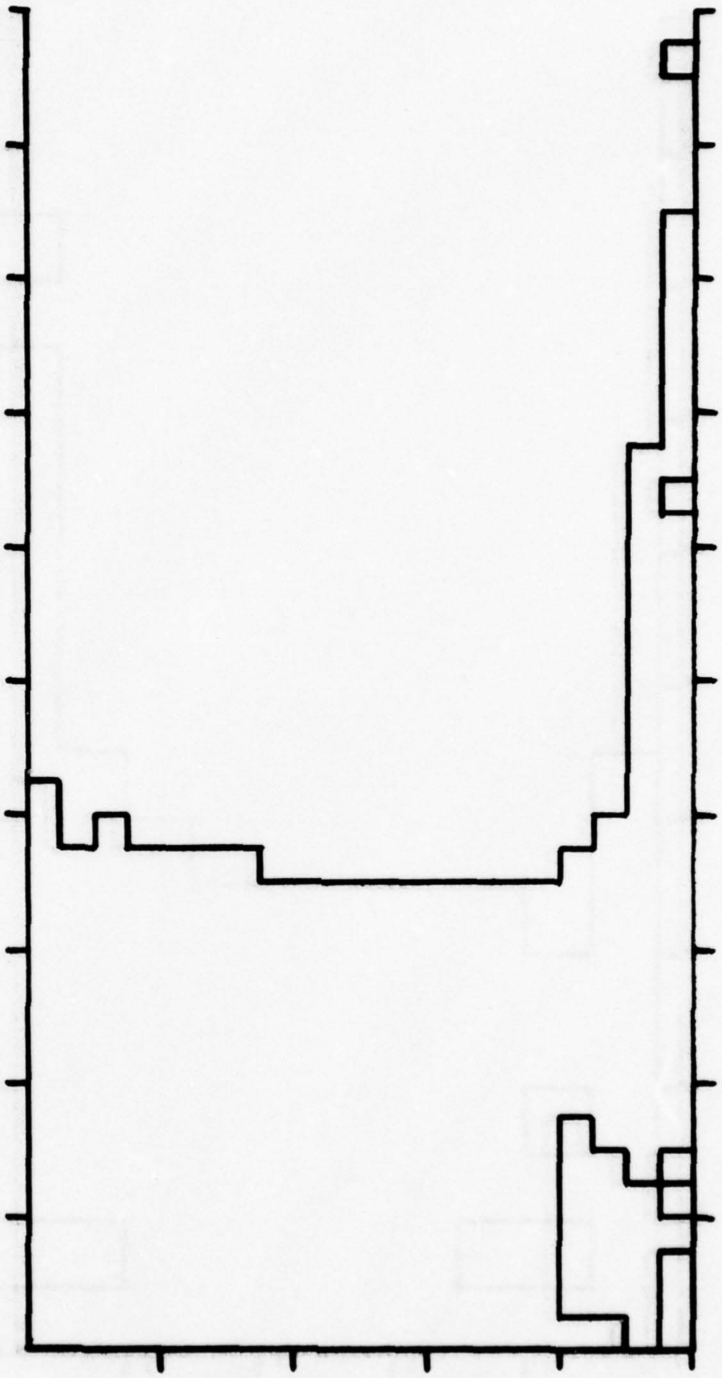


Fig. 19 Water surface for run 3 at time t = 0.6.

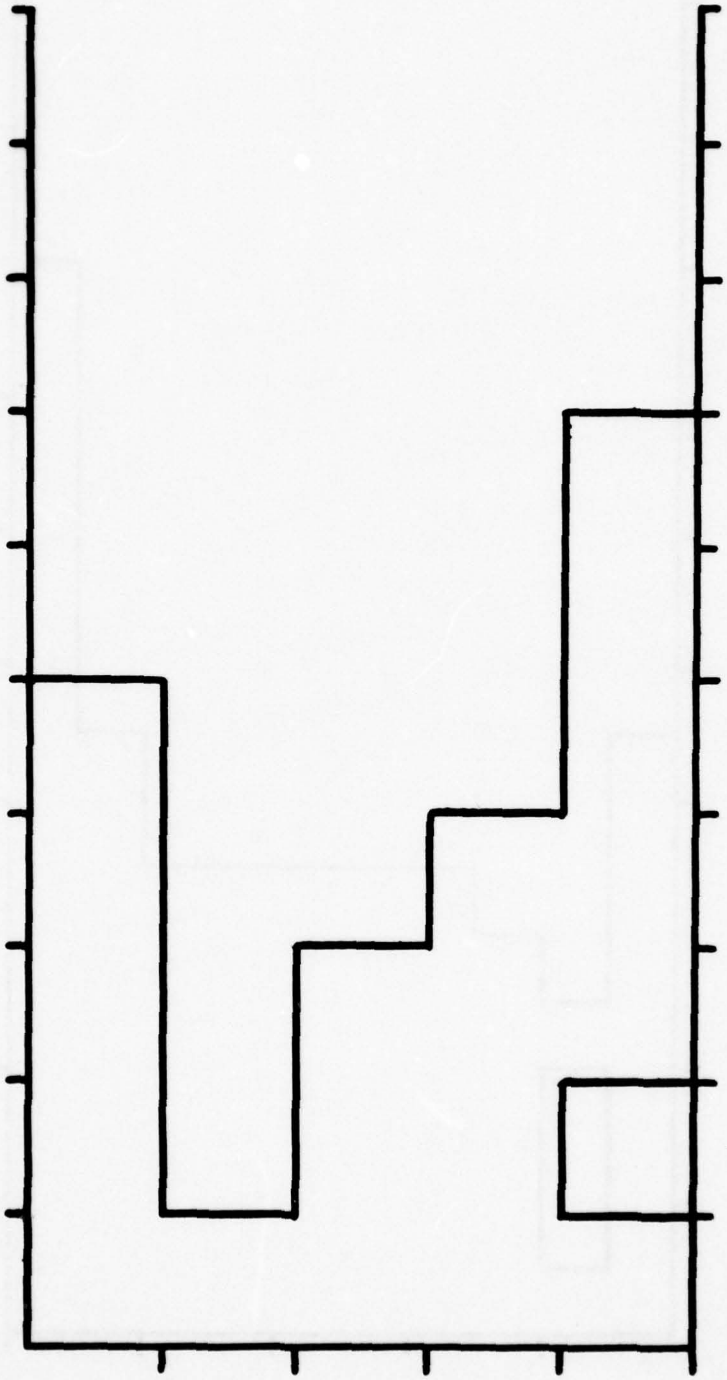


Fig. 20 Water surface for run 1 at time t = 0.7.

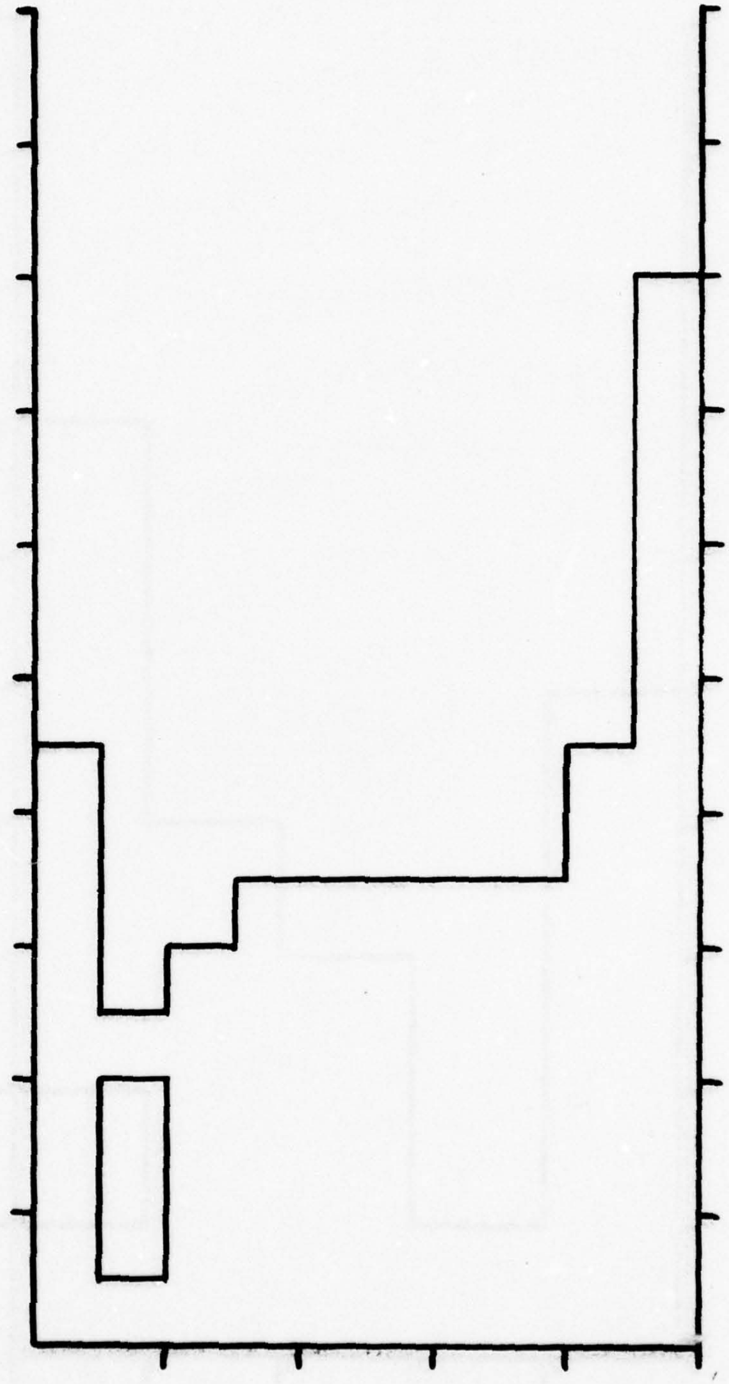


Fig. 21 Water surface for run 2 at time t = 0.7.

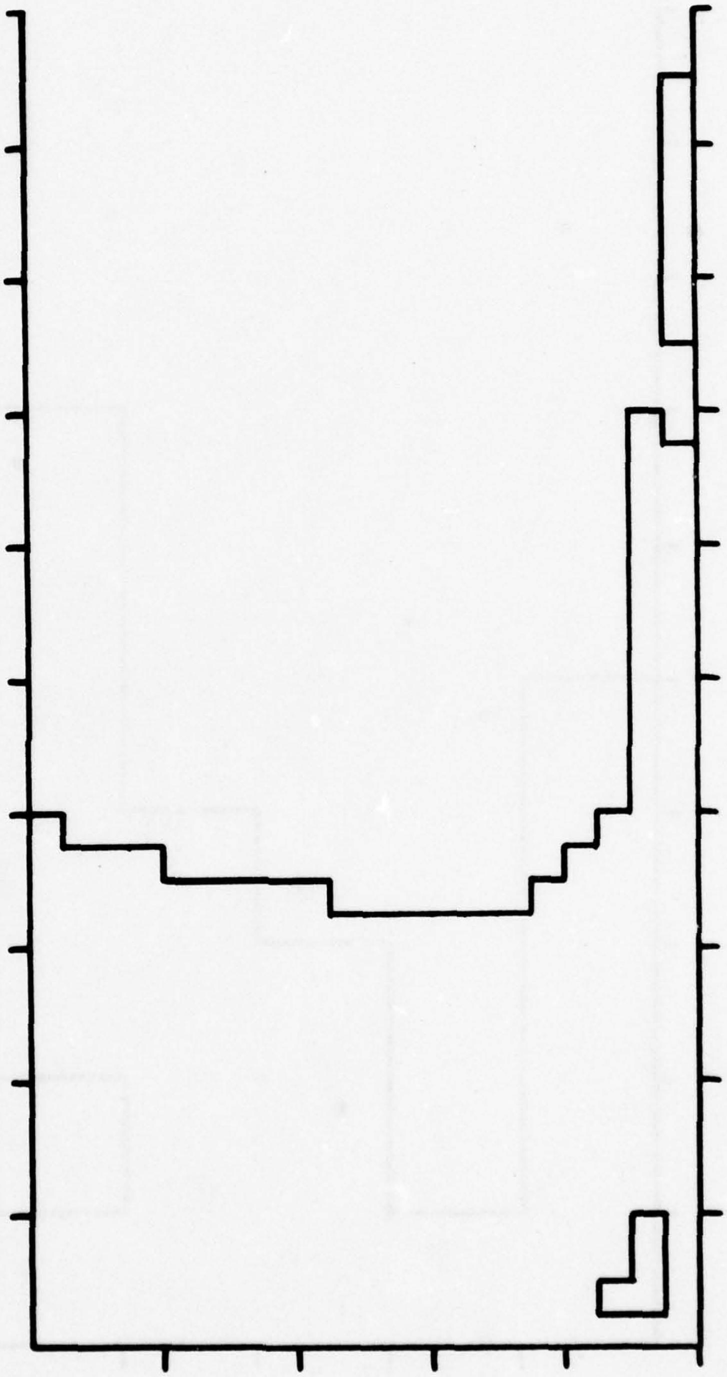


Fig. 22 Water surface for run 3 at time t = 0.7.

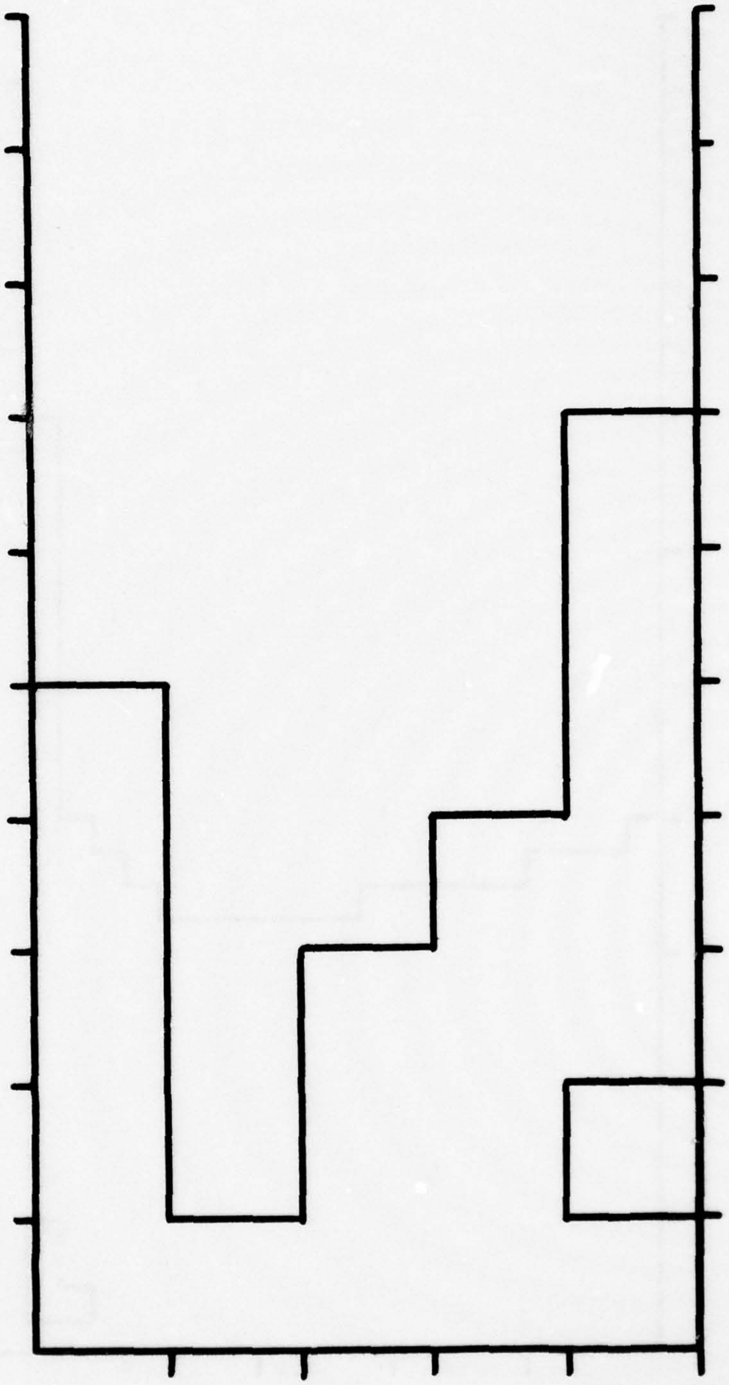


Fig. 23 Water surface for run 1 at time t = 0.8.

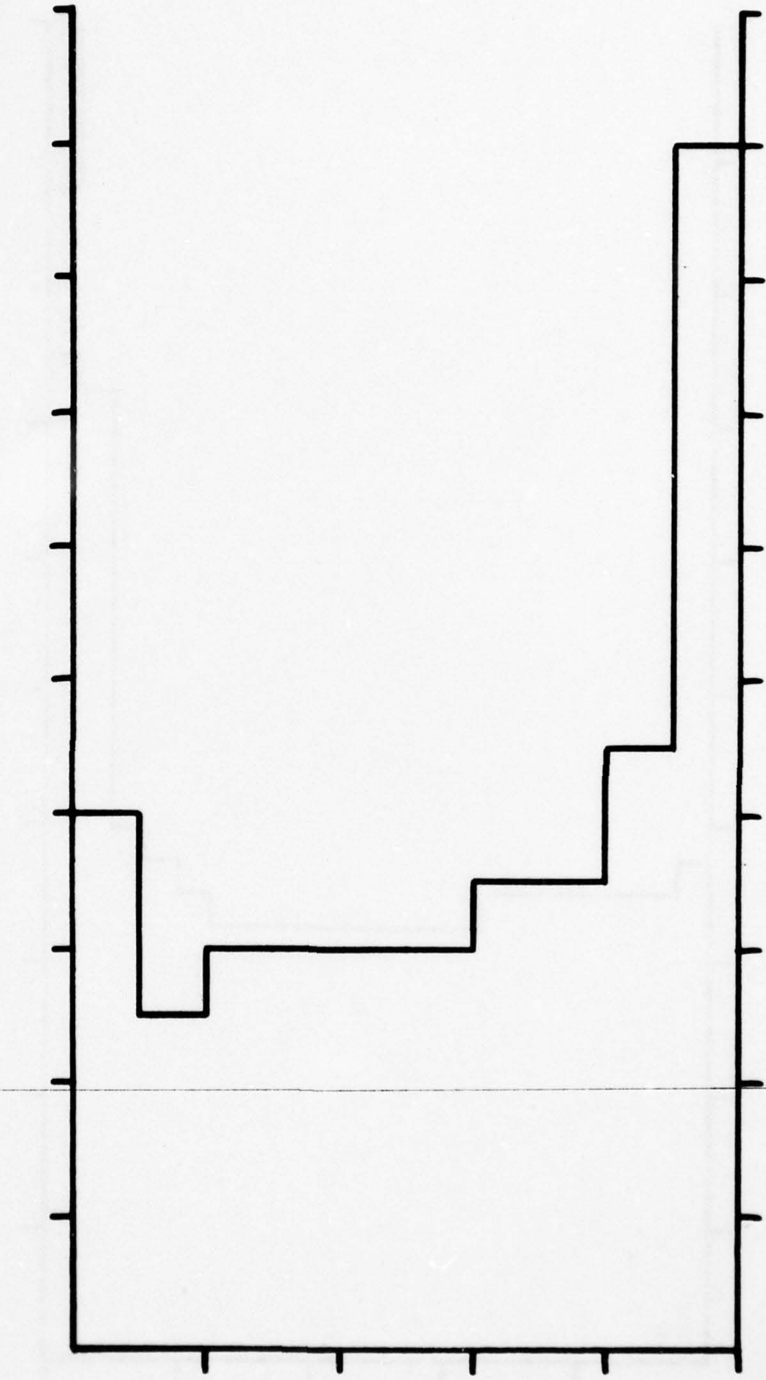


Fig. 24 Water surface for run 2 at time t = 0.8.

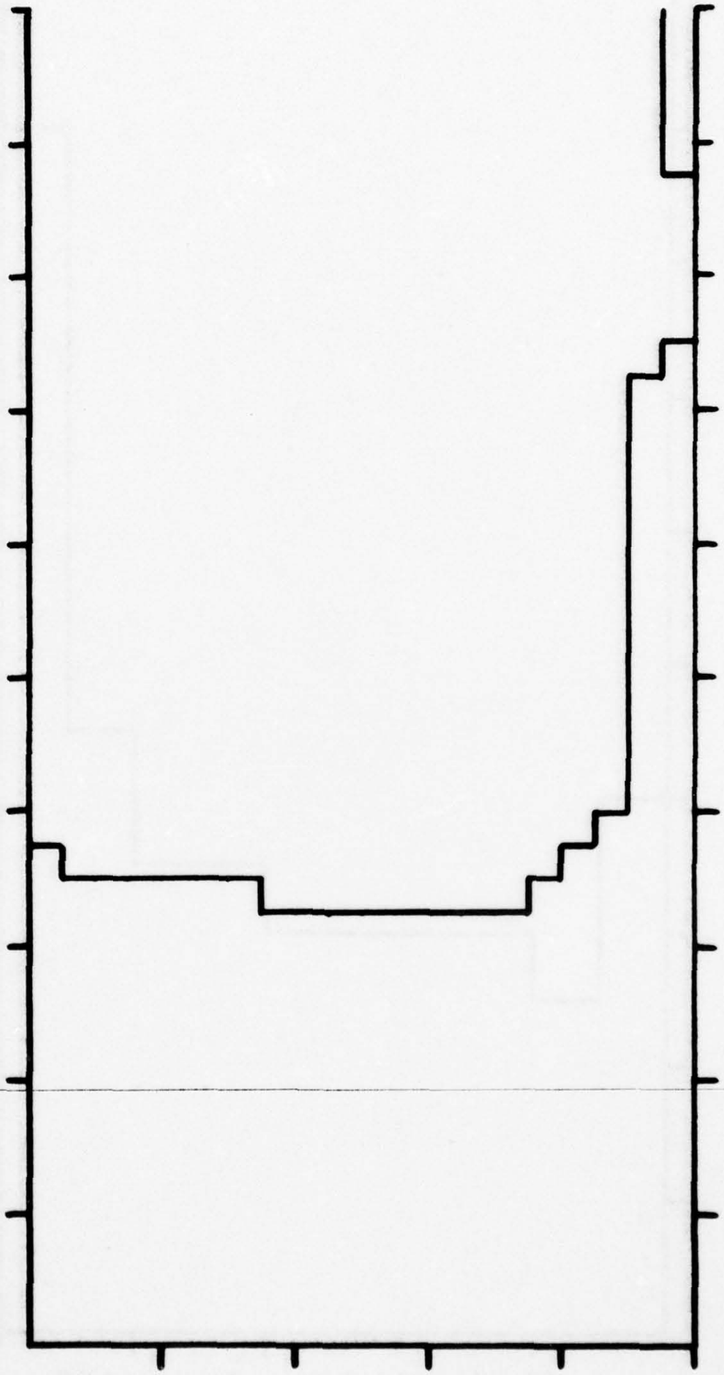


Fig. 25 Water surface for run 3 at time t = 0.8.

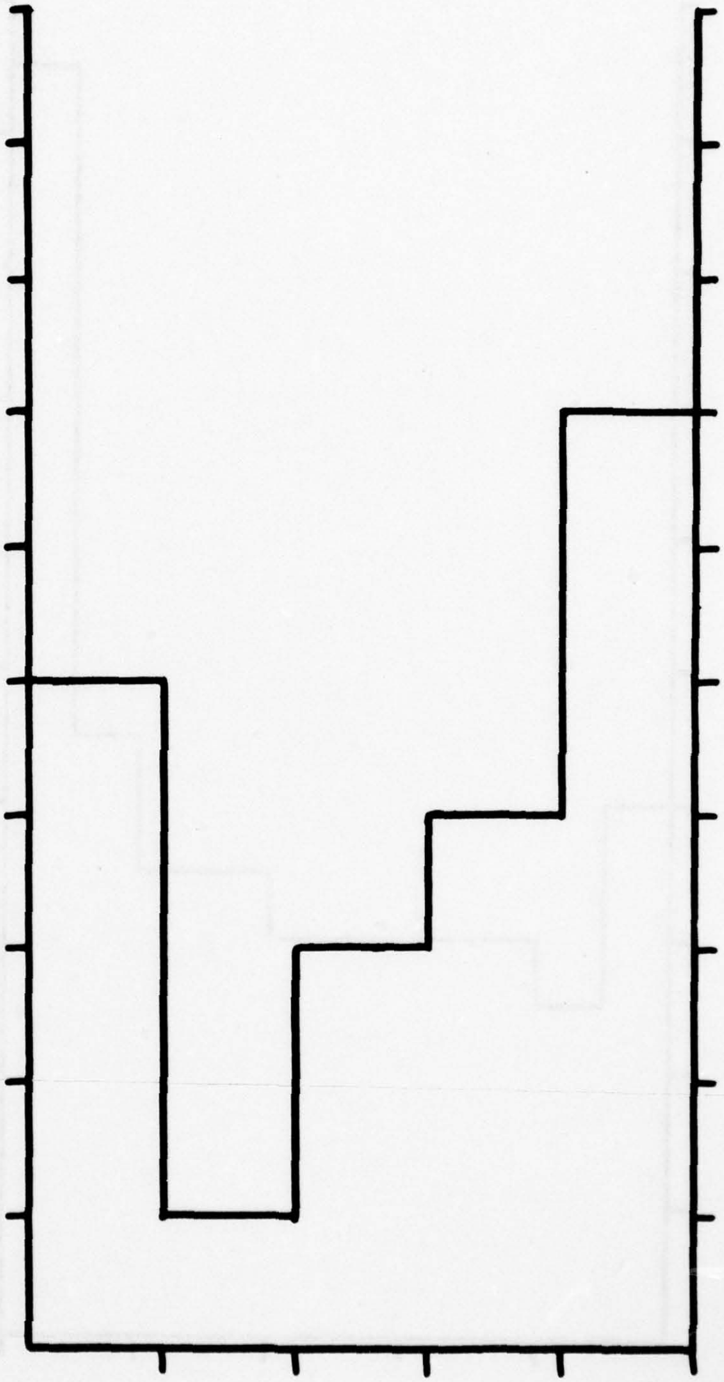


Fig. 26 Water surface for run 1 at time t = 0.9.

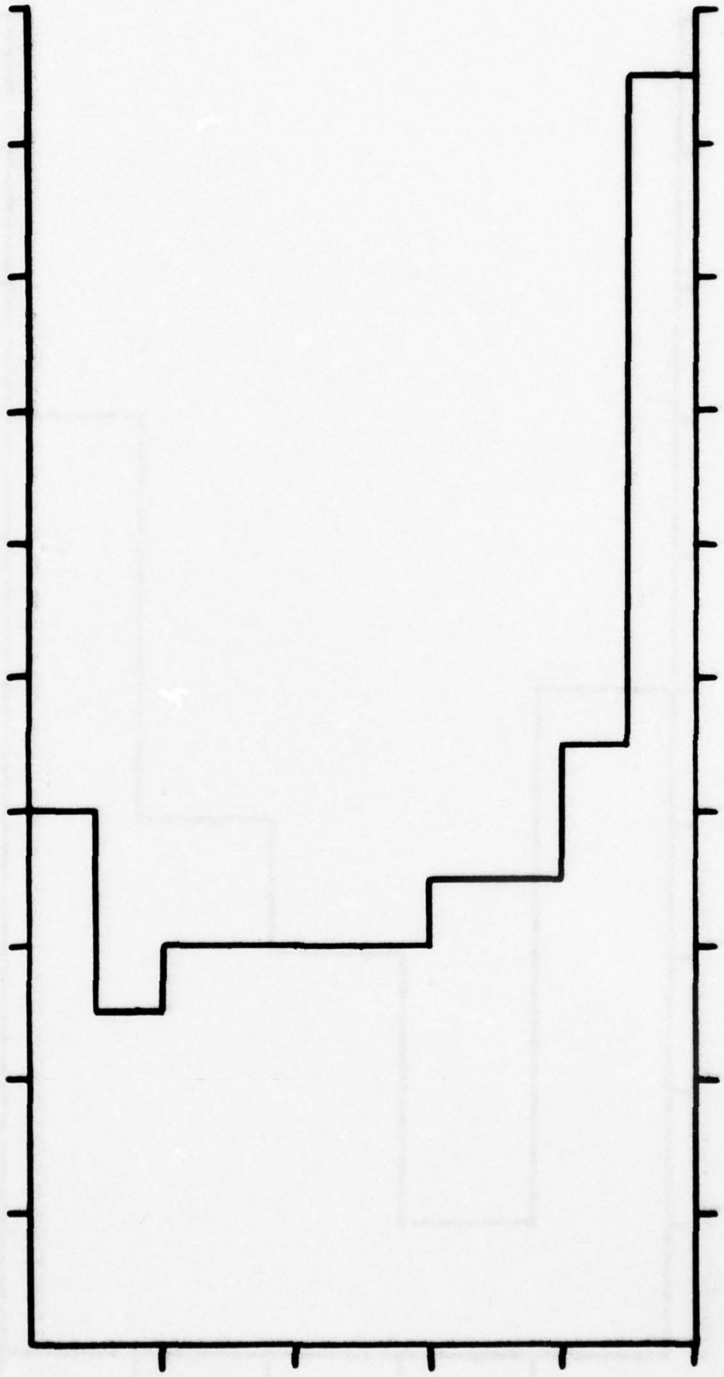


Fig. 27 Water surface for run 2 at time t = 0.9.

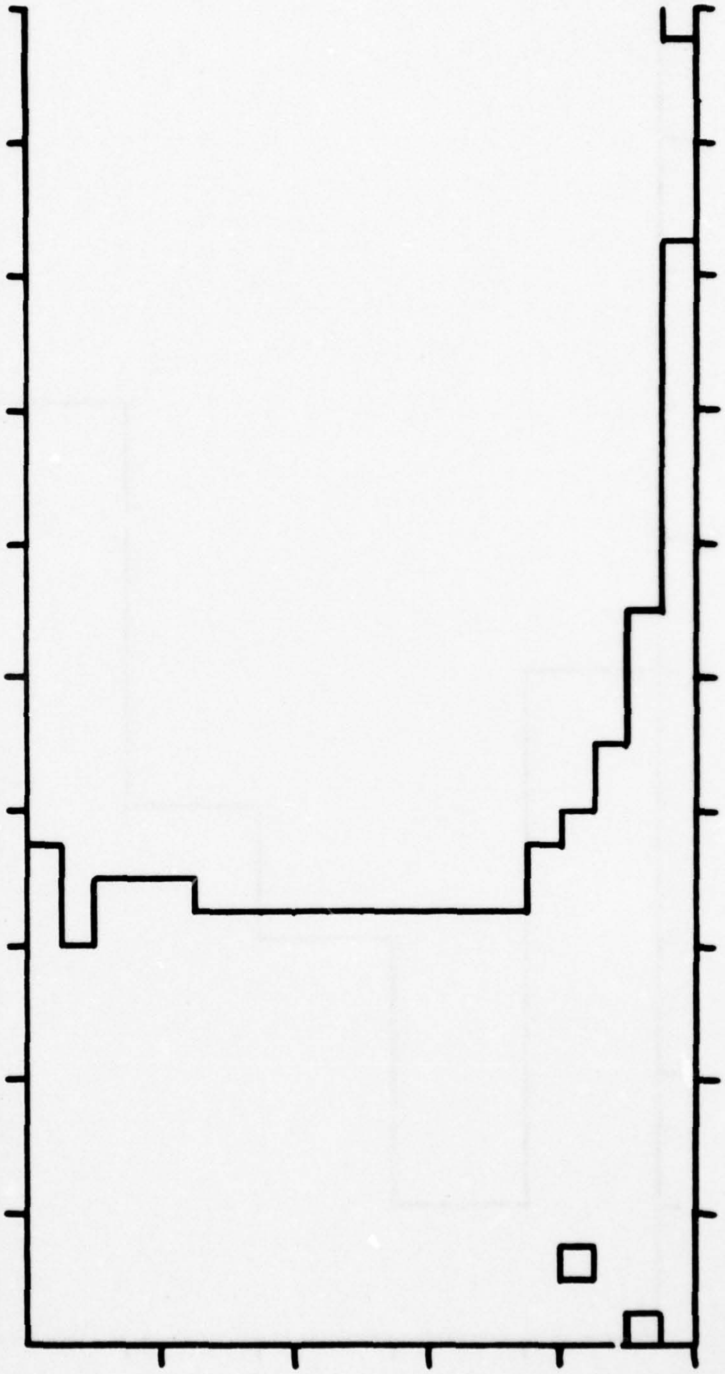


Fig. 28 Water surface for run 3 at time t = 0.9.

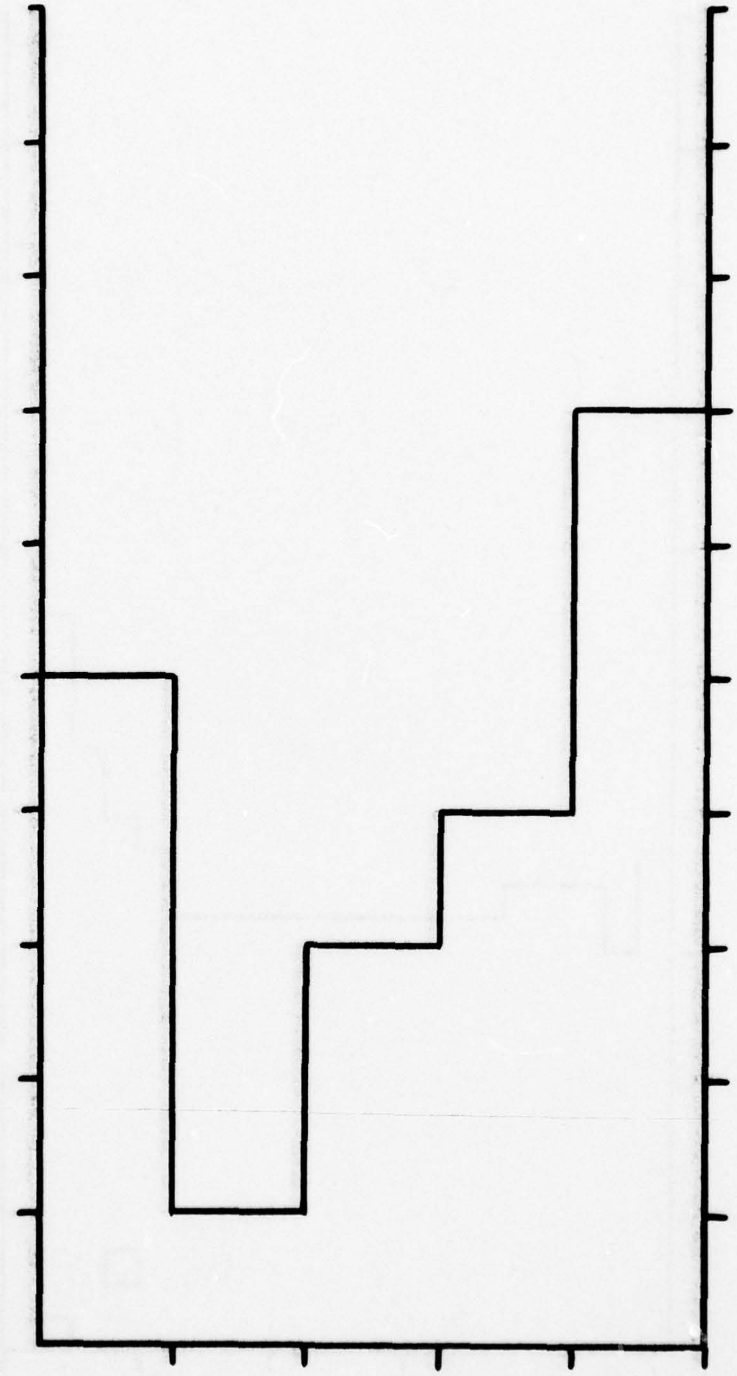


Fig. 29 Water surface for run 1 at time t = 1.0.

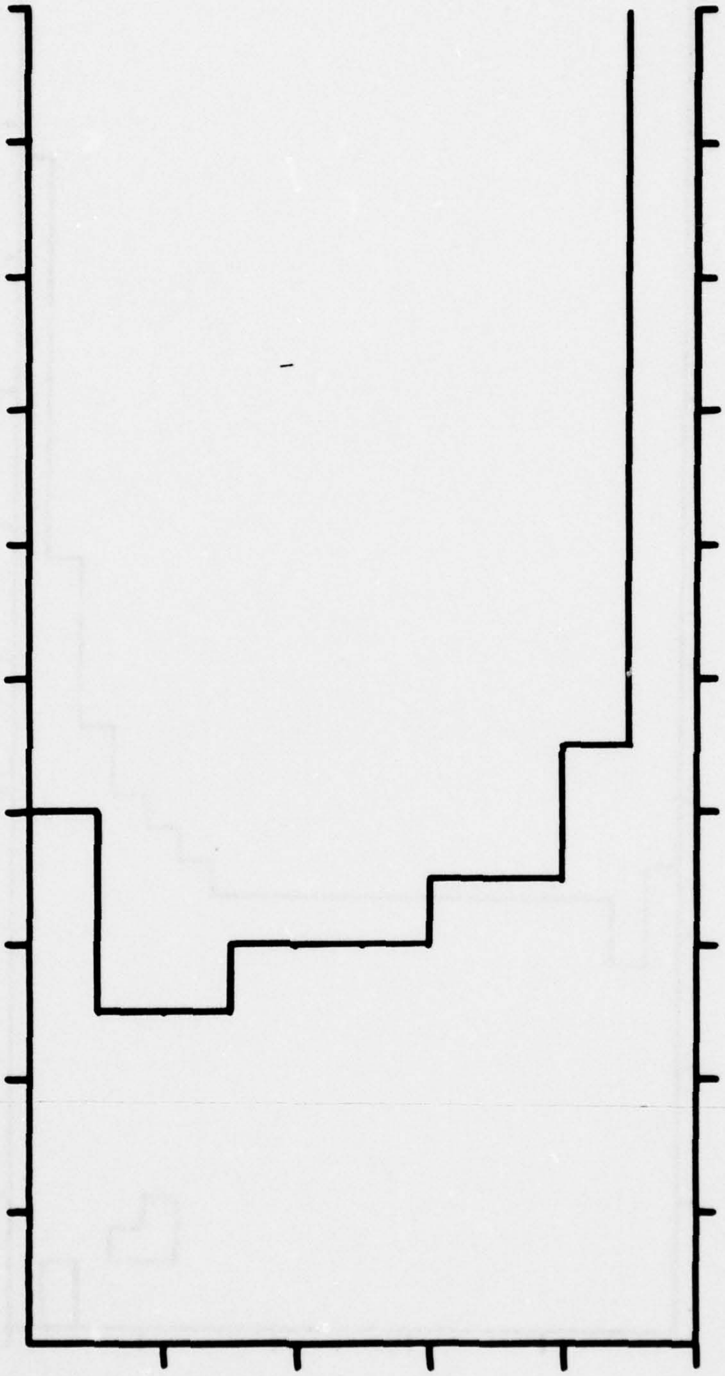


Fig. 30 Water surface for run 2 at time t = 1.0.

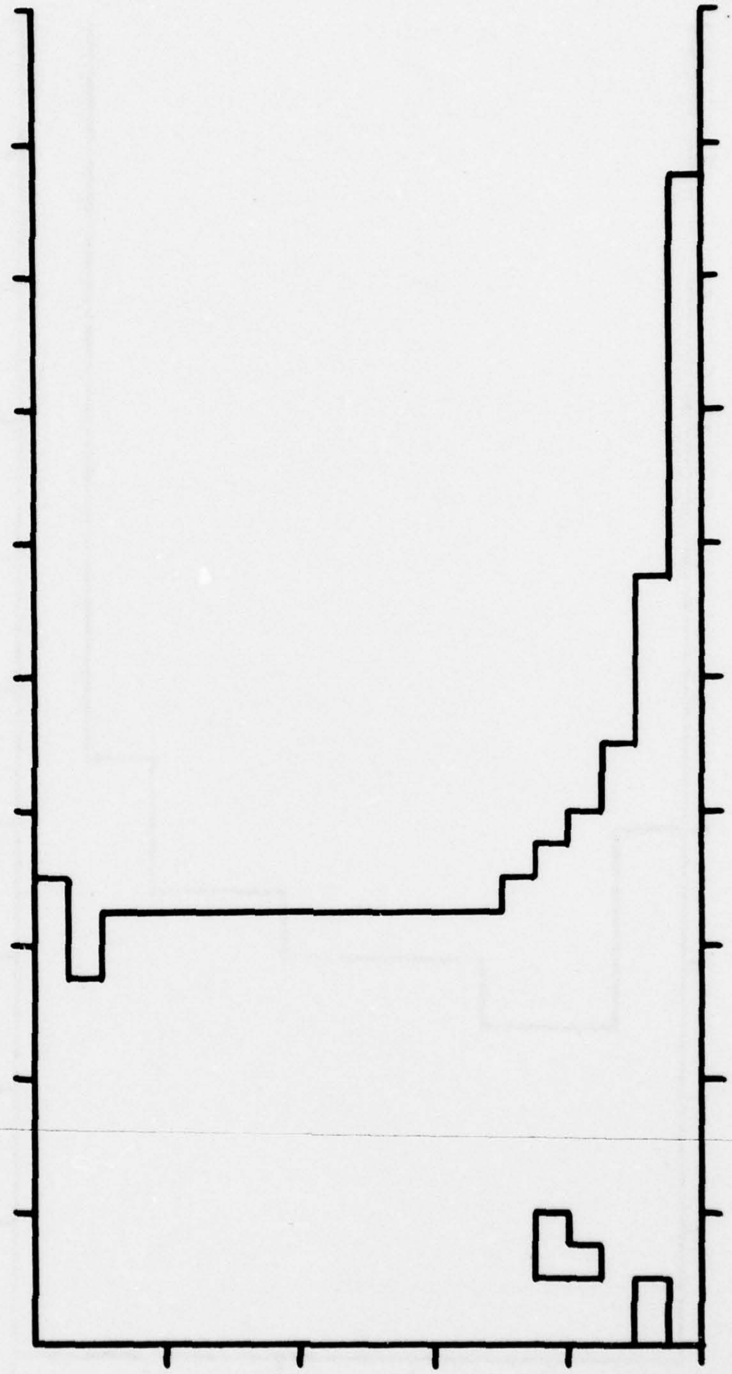


Fig. 31 Water surface for run 3 at time t = 1.0.

5. LIMITATIONS AND IMPROVEMENTS OF THE PROGRAM

One of the most serious limitations of the program is in the solution of the hyperbolic conservation laws (Eqs. 6 through 9). We observe from Eqs. 39 and 42 through 56 that our algorithm allows a diffusion of mass across the computational grid in addition to the convective flow described by the conservation laws, and this reduces the clarity of the delineation of the free surface. There are relatively simple remedies available at slight cost in computational effort. One such remedy is along the following lines: during the convective process, when we move parcels of fluid from one cell to another, we may associate with the fluid so moved not only a mass and momentum, but also a center of mass. Thus at each time we may assign a center of mass to the fluid in each cell. And we may consider the fluid in the cell i,j, with center of mass (x_{0ij}, z_{0ij}) to reside in a rectangle of area

$$A = 4 \min(x_{0ij} - x_{i-1/2}, x_{i+1/2} - x_{0ij}) \min(z_{0ij} - z_{j-1/2}, z_{j+1/2} - z_{0ij}),$$

unless the ratio of the mass in the cell, m_{ij}, to A exceeds ρ_0. In that case we may consider m_{ij} to be uniformly distributed over cell i,j with density $\frac{m_{ij}}{\Delta x_i \Delta z_j}$, as we have done heretofore. By such a procedure, we can limit the diffusion of mass due to the finite cell size. As a practical matter, we have found this spurious diffusion of mass to be greatest in the case where g = 0. When g > 0 we have observed, not unexpectedly, that the gravity tends to stabilize the free surface, which is usually confined to one or two computational cells in thickness.

The novelty of our approach to hydrodynamics lies in the replacement of the usual divergence condition on the velocity by the constraint $\rho \lesssim \rho_0$. That part that deals with the hyperbolic conservation laws is not new, at least from a computational point of view. It may be that other numerical work on such conservation laws is more satisfactory than our own treatment, and that problems such as the mass diffusion just referred to have already been adequately handled in other investigations. Work currently in progress by M. Y. Hussaini (Ref. 5) uses our treatment of the density constraint in a three-dimensional incompressible flow and solves the hyperbolic conservation laws using a MacCormack "higher-order" hyperbolic solver (Ref. 6).

In the examples reported in the last section, we noted the need for an improved treatment of the velocities at the rigid boundaries. Characteristically, we find a rather large outward normal velocity at the cells adjacent to the rigid boundary. This sort of behavior is encouraged by our numerical representation of Eq. 27 as Eq. 87. We may expect a more satisfactory treatment by regarding the right-hand side of Eq. 27 as an integral over all $x' \epsilon R^2$ and $\theta^n(x')$ extended symmetrically across the rigid boundary ∂D.

The determination of v can itself be made more efficient than the method used in Section 2, where a Stefan problem was solved until steady state was reached. For example, one may make use of the monotone dependence of the solution of the steady-state Stefan problem on $\tilde{\rho}$, and also of the fact that this solution may be obtained by solving a succession of N steady-state Stefan problems with initial data $\rho_i \geq 0$, $\sum_{i=1}^{N} \rho_i = \tilde{\rho}$ (Ref. 7), to obtain directly a lower approximation to v, with the remainder of v being determined iteratively. For example, this would be desirable in the solution of problems in water of great depth, where v, being proportional to the pressure, would get quite large.

Ref. 5. M. Y. Hussaini (private communication).

Ref. 6. R. W. MacCormack, "An Efficient Numerical Method for Solving the Time-Dependent Compressible Navier-Stokes Equations at High Reynolds Number," Comput. Appl. Math., Vol. 18, 1976, p. 49.

Ref. 7. J. C. W. Rogers, "Steady State of a Nonlinear Evolutionary Equation, Seminaires IRIA, Analyse et Contrôle de Systèmes, 1978.

An example of small improvements that might be made in the program is the following: at present, we only solve the constraint (Eq. 1) approximately, getting $\rho \leq \rho_0 + \varepsilon_1$, where ε_1 is given in Eq. 67. Thus we would expect the density computed in the liquid domain at each time to exceed ρ_0 by a small amount proportional to ε_1, and this in turn should lead to some "settling" of the liquid (in the direction of the gravitational force). This situation can be ameliorated by solving the Stefan problem with ρ_0 replaced by $\rho_0 - \frac{\varepsilon_1}{2}$ in the definition of the function f in Eq. 10b, and replacing the test (Eq. 67) by the test

$$\max_{\substack{1 \leq i \leq I \\ 1 \leq j \leq J}} (\rho^n_{ij} - \rho_0) \leq \frac{\varepsilon_1}{2} \quad .$$

For the future, the first thing we would like to do is to improve the treatment of velocities at the rigid boundary, especially the numerical representation of Eq. 27. Beyond that, we are thinking of making the code applicable to the computation of internal waves in stratified fluids. This would require only a relatively modest addition to the program as it now stands (Ref. 1).

ACKNOWLEDGMENT

This work has been supported by the Office of Naval Research under Task No. NR 334-003. Some of the calculations have been done under Contract N00024-78-C-5384 with the Naval Sea Systems Command.

REFERENCES

1. J. C. W. Rogers, "Incompressible Flows as a System of Conservation Laws with a Constraint," Seminaires IRIA, Analyse et Contrôle de Systèmes, 1978.

2. J. C. W. Rogers, "Stability, Energy Conservation, and Turbulence for Water Waves," Seminaires IRIA, Analyse et Contrôle de Systèmes, 1978.

3. H. Brezis, A. E. Berger, and J. C. W. Rogers, "A Numerical Method for Solving the Problem $u_t - \Delta f(u) = 0$" (to be published).

4. J. C. W. Rogers, "An Algorithm for a Hyperbolic Free Boundary Problem," APL/JHU TG 1309, May 1977.

5. M. Y. Hussaini (private communication).

6. R. W. MacCormack, "An Efficient Numerical Method for Solving the Time-Dependent Compressible Navier-Stokes Equations at High Reynolds Number," Comput. Appl. Math., Vol. 18, 1976, p. 49.

7. J. C. W. Rogers, "Steady State of a Nonlinear Evolutionary Equation," Seminaires IRIA, Analyse et Contrôle de Systèmes, 1978.

8. "The Frank T. McClure Computing Center User's Guide," APL/JHU BCS-1-92, 1 Sep 1978.

Appendix A
PROGRAM DESCRIPTION AND LISTING

The following water wave program was written for the optimizer and checkout PL/I compilers and executed on an IBM 360/91 computer at the Frank T. McClure Computing Center of APL (Ref. 8).

Originally, the program was written as one long program, but we found that initial conditions were easier to program in-line, rather than read in as input data, so the program was broken into various sections.

The main procedure first states various constants for a given run. It also tests certain conditions for convergence, when to stop and when to print answers, and when to write on a disk in order to restart or continue a problem at a future time.

Procedure INITAL computes the x- and z-coordinates of points in the extended computational grid.

Procedure PSAQS computes the matrix elements that simulate the effect of diffusion in the x- and z-directions by transforming quantities of mass and momentum in each computational cell into new values through multiplication by the appropriate matrices.

Procedure MASMON computes the effect of convection for a time step on the values of mass and momentum in each computational cell.

Procedure DENSTY computes a new set of masses for each computational cell at the end of each time step by satisfying the constraint on the density.

Procedure MOMEN computes the final amounts of momentum in each computational cell at the end of each time step.

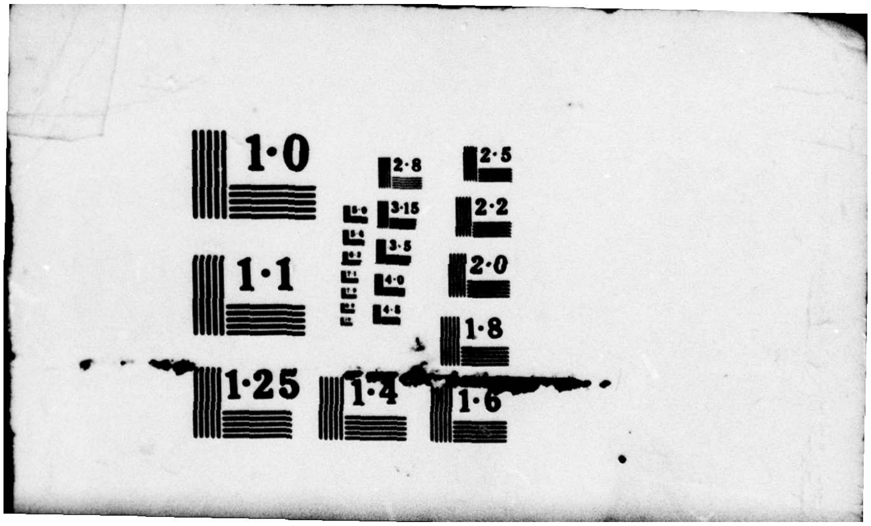
Procedure PRNTAL does what its name implies — it prints out the desired information.

Ref. 8. "The Frank T. McClure Computing Center User's Guide," APL/JHU BCS-1-92, 1 Sep 1978.

AD-A072 944   JOHNS HOPKINS UNIV LAUREL MD APPLIED PHYSICS LAB   F/G 20/4
COMPUTATION OF WATER WAVES.(U)
JUL 79 J C ROGERS, S FAVIN   N00024-78-C-5384
UNCLASSIFIED   APL/JHU/TG-1325   NL

2 OF 2
AD
A072944

END
DATE
FILMED
9 79
DDC


1·0
2·8
2·5
3·15
2·2
3·5
1·1
2·0
4·0
4·5
1·8
1·25
1·4
1·6

EXCEPT WHERE NOTED, ALL VARIABLES ARE SINGLE PRECISION FLOAT BINARY NUMBERS.

THE EXCEPTION TO THE RULE ARE VARIABLES I,IMAX,IM1,IM2,.... WHICH FOLLOW THE NORMAL NAMING CONVENTIONS.

IF A VARIABLE IS DIMENSIONED, IT WILL BE NOTED BY PARENTHESES.

VARIABLES	WHERE USED	DESCRIPTION
DA	MAIN, PSAQS, DENSTY, MOMEN	STEP SIZE OF INDEPENDENT VARIABLE OF EQUATION 10 A, LABELED "$\Delta\alpha$" IN THE TEXT.
DEBUG	MAIN, DENSTY MOMEN	BIT(1); A TRUTH/FALSE SWITCH TO PRODUCE DEBUG OUTPUT.
DG	MOMEN	STEP SIZE OF INDEPENDENT VARIABLE γ OF EQUATION 22 A.
DT	MAIN, MASMON, DENSTY, MOMEN	TIME STEP, LABELED AS "τ" IN THE TEXT.
DX(20)	MAIN, PSAQS, INITAL, MASMON, DENSTY, MOMEN	WIDTHS OF CELLS R_{ij} IN EQUATION 36.
DZ(40)	MAIN, PSAQS, INITAL, MASMON, DENSTY, MOMEN	HEIGHTS OF CELLS R_{ij} IN EQUATION 36.
EPS	MAIN, PRNTAL, MASMON	SMALL CUT-OFF TO KEEP FROM DIVIDING BY ZERO IN EQUATION 38.
EPS1	MAIN, DENSTY	SMALL PARAMETER WHICH DETERMINES WHEN DENSITY CONSTRAINT EQUATION 81 IS SATISFIED WITH SUFFICIENT ACCURACY.
EPS2	MAIN	SMALL PARAMETER INTRODUCED IN EQUATION 89, WHICH DETERMINES WHETHER LAST TERM ON RIGHT-HAND SIDE OF EQUATION 13A IS SIGNIFICANT.
ERR	MAIN, MASMON, DENSTY	BIT(1); A TRUTH/FALSE SWITCH TO TELL THE MAIN PROGRAM IF CERTAIN CONVERGENCE WAS MET.
G	MAIN, PRNTAL, MASMON	GRAVITATIONAL CONSTANT, OCCURRING IN EQUATION 6.
GMAX	MAIN, MOMEN	CUT-OFF PARAMTER, LABELED AS "γ_0" IN EQUATION 95, FOR SOLUTION OF EQUATION 13A.

VARIABLES	WHERE USED	DESCRIPTION
IMAX	MAIN, PSAQS, INITAL, PRNTAL, MASMON, DENSTY, MOMEN	NUMBER OF CELLS INTO WHICH COMPUTATIONAL GRID IS DIVIDED IN X-DIRECTION.
IM1	MAIN, INITAL	= IMAX + 1
IM2	MAIN, INITAL	= IMAX + 2
IM21	MAIN, INITAL	= 2*IMAX + 1
IM22	MAIN, INITAL	= 2*IMAX + 2
IPL	MAIN, PRNTAL, MOMEN	COUNTER
ISL	MAIN, PRNTAL, DENSTY	COUNTER
I2	MAIN, INITAL, MASMON	= 2*IMAX
JMAX	MAIN, PSAQS, INITAL, PRNTAL, MASMON, DENSTY, MOMEN	NUMBER OF CELLS INTO WHICH COMPUTATIONAL GRID IS DIVIDED IN Z-DIRECTION.
JM1	MAIN, INITAL	= JMAX + 1
JM2	MAIN, INITAL	= JMAX + 2
JM21	MAIN, INITAL	= 2*JMAX + 1
JM22	MAIN, INITAL	= 2*JMAX + 2
J2	MAIN, INITAL, MASMON	= 2*JMAX
M(20,40)	MAIN, PRNTAL, MASMON, DENSTY	MASS IN EACH CELL R_{ij}
MOMX(20,40)	MAIN, PRNTAL, MASMON, MOMEN	X-COMPONENT OF MOMENTUM IN EACH CELL R_{ij}
MOMZ(20,40)	MAIN, PRNTAL, MASMON, MOMEN	Z-COMPONENT OF MOMENTUM IN EACH CELL R_{ij}
M2VX(20,40)	MAIN, DENSTY, MOMEN	CORRECTION TO X-COMPONENT OF MOMENTUM DUE TO DENSITY CONSTRAINT, LABELED "$(\Delta\mu)_x$" IN EQUATION 87A.
M2VZ(20,40)	MAIN, DENSTY, MOMEN	CORRECTION TO Z-COMPONENT OF MOMENTUM DUE TO DENSITY CONSTRAINT, LABELED "$(\Delta\mu)_z$" IN EQUATION 87B.
N	MAIN, PRNTAL	COUNTER
NMAX	MAIN	MAXIMUM FOR THE N COUNTER.
ONE	MAIN, MOMEN	THE VALUE 1.0
P(20,40)	MAIN, PSAQS, DENSTY, MOMEN	COEFFICIENTS GIVING EFFECT OF DIFFUSION IN X-DIRECTION OF MASS AND Z-COMPONENT OF MOMENTUM, GIVEN BY EQUATIONS 65 AND 75.

VARIABLES	WHERE USED	DESCRIPTION
PCON	MAIN, PSAQS	$\sqrt{\Delta\alpha/\pi}$, WHERE $\Delta\alpha$ IS STEP SIZE OF INDEPENDENT VARIABLE α IN EQUATION 10A.
P1(20,40)	PSAQS, MOMEN	COEFFICIENTS GIVING EFFECT OF DIFFUSION IN X-DIRECTION OF X-COMPONENT OF MOMENTUM, GIVEN BY EQUATIONS 98 AND 100.
Q(40,40)	PSAQS, DENSTY	COEFFICIENTS GIVING EFFECT OF DIFFUSION IN Z-DIRECTION OF MASS AND X-COMPONENT OF MOMENTUM, GIVEN BY EQUATIONS 69 AND 78.
Q1(40,40)	PSAQS, MOMEN	COEFFICIENTS GIVING EFFECT OF DIFFUSION IN Z-DIRECTION OF Z-COMPONENT OF MOMENTUM, GIVEN BY EQUATIONS 99 AND 101.
RHO(20,40)	MAIN, MASMON, DENSTY	DENSITY IN CELL R_{ij}, GIVEN BY EQUATION 80.
RHOC	MAIN, DENSTY, MOMEN	CHARACTERISTIC DENSITY OF FLUID, LABELED "ρ_0" IN TEXT.
SQDA	MAIN, PSAQS	$\sqrt{\Delta\alpha}$, WHERE $\Delta\alpha$ IS STEP SIZE OF INDEPENDENT VARIABLE α IN EQUATION 10A.
TM(40,80)	MASMON	MASS IN EACH CELL OF EATENDED GRID AFTER CONVECTION, DENOTED BY $m^*_{k\ell}$ IN EQUATION 53A.
TMOMX(40,80)	MASMON, MOMEN	X-COMPONENT OF MOMENTUM IN EACH CELL OF EXTENDED GRID AFTER CONVECTION, DENOTED BY $\mu^*_{x\,k\ell}$ IN EQUATION 53B.
TMOMZ(40,80)	MASMON, MOMEN	Z-COMPONENT OF MOMENTUM IN EACH CELL OF EXTENDED GRID AFTER CONVECTION, DENOTED BY $\mu^*_{z\,k\ell}$ IN EQUATION 53C.
TPCON	MAIN, PSAQS	$2\sqrt{\Delta\alpha/\pi}$, WHERE $\Delta\alpha$ IS STEP SIZE OF INDEPENDENT VARIABLE α IN EQUATION 10A.
TSDA	MAIN, PSAQS	$2\sqrt{\Delta\alpha}$, WHERE $\Delta\alpha$ IS STEP SIZE OF INDEPENDENT VARIABLE α IN EQUATION 10A.
TWO	MAIN, PSAQS, PRNTAL, DENSTY	THE VALUE 2.0
U(20,40)	MAIN, PRNTAL, MASMON, MOMEN	X-COMPONENT OF VELOCITY IN CELL R_{ij} GIVEN BY EQUATION 38A.

VARIABLES	WHERE USED	DESCRIPTION
V(20,40)	PRNTAL, DENSTY, MOMEN	QUANTITY WHICH DESCRIBES EFFECT OF DENSITY CONSTRAINT UN MOMENTUM GIVEN BY EQUATION 26.
VMAX	MAIN, DENSTY, MOMEN	MAXIMUM UF V_{ij} OVER CELLS R_{ij}, LABELED " V^{+} " IN EQUATION 91.
V1(20,40)	DENSTY, MOMEN	SCALED VALUES OF V_{ij} TU INSURE STABILITY OF ALGORITHM, GIVEN BY EQUATION 94.
W(20,40)	MAIN, PRNTAL, MASMON, MOMEN	Z-COMPONENT OF VELOCITY IN CELL R_{ij} GIVEN BY EQUATION 38B.
XMH(81)	PSAQS, INITAL, MASMON, DENSTY	X-COORDINATES OF LEFT-HAND SIDES OF CELLS IN EXTENDED COMPUTATIONAL GRID, GIVEN BY EQUATIONS 35B AND 43A.
XPH(60)	PSAQS, INITAL, MASMON, DENSTY	X-COURDINATES OF RIGHT-HAND SIDES OF CELLS IN EXTENDED COMPUTATIONAL GRID, GIVEN BY EQUATIONS 35B AND 43A.
ZMH(81)	PSAQS, INITAL, PRNTAL, MASMON, DENSTY	Z-COORDINATES OF BOTTOMS OF CELLS IN EXTENDED COMPUTATIONAL GRID, GIVEN BY EQUATIONS 35B AND 43B.
ZO	MAIN, PRNTAL, DENSTY, MOMEN	THE VALUE 0.
ZPH(80)	PSAQS, INITAL, PRNTAL, MASMON, DENSTY	Z-COORDINATES OF TUPS OF CELLS IN EATENDED COMPUTATIONAL GRID, GIVEN BY EQUATIONS 35B AND 43B.

```
                                   PAGE  1
ROGERS:    PROC  OPTIONS(MAIN);   /*   WATER WAVES  6/25/78                */

        DCL (ATAN,SQRT) BUILTIN;
        DCL (INITAL, PRNTAL, PSAQS, MASMON, DENSTY, MOMEN) ENTRY;

        DCL DISK1 FILE SEQUENTIAL RECORD;
        DCL DISK2 FILE SEQUENTIAL RECORD;

        DCL (Q,Q1) (40,40) FLOAT BIN EXT;
        DCL (P,P1,       M,MOMX,MOMZ,RHO,U,V,W,  V1   )(20,40) FLOAT BIN EXT;
        DCL (M2VX, M2VZ)(20,40) FLOAT BIN EXT;
        DCL (TMOMX, TMOMZ, TM) (40,80) FLOAT BIN EXT;
        DCL (DX(20), DZ(40), XMH(81),ZMH(81), XPH(60),ZPH(80))
                                                   FLOAT BIN EXT;

        DCL (DA, DG, DT, EPS, EPS1, EPS2, G, GMAX, PCON, RHOC, SQDA,
           TPCON, TSDA, VMAX) FLOAT BIN EXT;
        DCL (ZO, ONE, TWO, PI) FLOAT BIN EXT;

        DCL (IMAX,JMAX,ISL,IPL,N,IM1,IM2,IM21,IM22,I2,JM1,JM2,JM21,JM22,
           J2) FIXED BIN(31) EXT;
        DCL (I,J,NMAX) FIXED BIN(31);
        DCL (DEBUG, ERR) BIT(1) EXT;

        ON UNDERFLOW;
/**
        DEBUG = '1'B;
**/
        DEBUG = '0'B;
        ERR = '0'B;
        ISL, IPL = 0;
        ZO = 0.0;
        ONE = 1.0;
        TWO = 2.0;
        PI = 4.0*ATAN( ONE );

/*      -----------  INPUT CASE 7/6/78           --------------------      */
        NMAX =  20;
        IMAX = 10;
        JMAX = 20;
        N=0;
        G=ONE;
        EPS = 1.0E-5;
        EPS1 = 1.0E-2;
        EPS2 = 1.0E-3;
```

```
                                        PAGE  2
         DA = 0.05E0;
         RHOC = 1.0;
         GMAX = 10.0;
         SQDA = SQRT( DA );
         TSDA = 2.0 * SQDA;
         PCON = SQRT( DA/PI );
         TPCON = TWO * PCON;

         IM1 = IMAX + 1;
         IM2 = IMAX + 2;
         I2 = 2*IMAX;
         IM22 = I2+2;
         IM21 = I2 + 1;
         JM1 = JMAX + 1;
         JM2 = JMAX + 2;
         J2 = 2*JMAX;
         JM21 = J2 + 1;
         JM22 = J2 + 2;

      /*  --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- */


         DT = 0.05E0;

              MOMX,MOMZ,M,RHO,U,W = Z0;


         DX = 0.5E0;
         DZ = 0.5E0;
         DO I=1 TO IMAX;
         DO J=1 TO 4;
         M(I,J) = 0.25E0;
         END;   END;
         DO I=1 TO 4;
         DO J=1 TO 14;
         M(I,J) = 0.25E0;
         END;   END;
         DO I=1 TO IMAX;
         DO J=1 TO 4;
         MOMX(I,J) = -10.0*M(I,J);
         END;   END;
         DO I=1 TO 4;
         DO J=5 TO 14;
         MOMZ(I,J) = -10.0*M(I,J);
         END;   END;
```

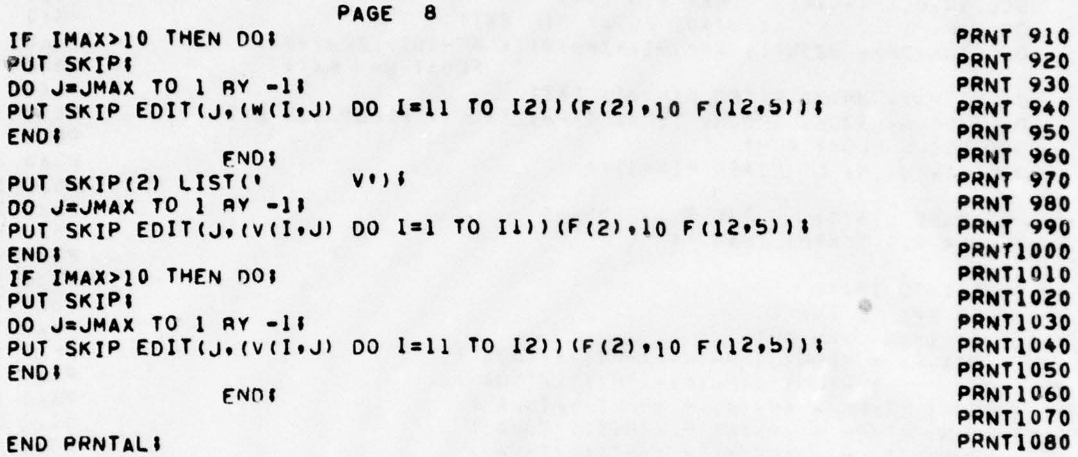
```
                                 PAGE  3

        CALL INITAL;

        CALL PRNTAL;

 /*     --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- */
        CALL PSAQS;

 /*************
        IF ONE = 1.0 THEN GO TO FINI;
 *******/
 /*     --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- */


 /*     --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- */
 NEXTIME:
        N = N+1;
        IF N > NMAX THEN GO TO FINI;
        CALL MASMON;
        IF ERR THEN GO TO ERROUT;
        CALL DENSTY;
        IF ERR THEN GO TO ERROUT;
        IF VMAX <= EPS2 THEN DO;
            DO I=1 TO IMAX;
            DO J=1 TO JMAX;
            MOMX(I,J) = MOMX(I,J) + M2VX(I,J) / DT;
            MOMZ(I,J) = MOMZ(I,J) + M2VZ(I,J) / DT;
            END;   END;
            GO TO OUT;
                                              END;
        CALL MOMEN;
 /*     --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- */
 OUT:
     /*
        IF   MOD(N,10)=0   THEN
        IF   MOD(N, 2)=0   THEN
        IF   MOD(N, 5)=0   THEN
     */
        CALL PRNTAL;
```

```
                                        PAGE  4
          IF N=1 THEN DO;
          WRITE FILE(DISK2) FROM(M);
          WRITE FILE(DISK2) FROM(MOMX);
          WRITE FILE(DISK2) FROM(MOMZ);
          CLOSE FILE(DISK2);
                   END;
          IF N<NMAX THEN GO TO  NEXTIME;

          WRITE FILE(DISK2) FROM(M);
          WRITE FILE(DISK2) FROM(MOMX);
          WRITE FILE(DISK2) FROM(MOMZ);
          CLOSE FILE(DISK2);

          GO TO FINI;

 ERROUT:
          PUT SKIP LIST('HELP');
          CALL PRNTAL;
 FINI:
          END ROGERS;
//G.SYSPRINT DD OUTLIM=50000
//G.DISK2 DD DSN=RCP.FAV.HOGD3,DISP=OLD,
//       DCB=(RECFM=F,LRECL=3200,BLKSIZE=3200),SPACE=(3200,(6,2),RLSE)
```

```
                                PAGE  5
INITAL: PROC;

    DCL (DX(80), DZ(40), XMH(81),ZMH(81), XPH(60),ZPH(80))
                                          FLOAT BIN EXT;
    DCL (IMAX,JMAX,IM1,IM2,IM21,IM22,I2,JM1,JM2,JM21,JM22,
       J2) FIXED BIN(31) EXT;
    DCL (I, J) FIXED BIN(31);

    XMH(1)=0;
    DO I=1 TO IMAX;
    XMH(I+1) = XMH(I) + DX(I);
    END;
    DO I=IM2 TO IM21;
    XMH(I) = 2*XMH(IM1) - XMH(IM22-I);
    END;
    DO I=1 TO I2;
    XPH(I) = XMH(I+1);
    END;

    ZMH(1)=0;
    DO J=1 TO JMAX;
    ZMH(J+1) = ZMH(J) + DZ(J);
    END;
    DO J=JM2 TO JM21;
    ZMH(J) = 2*ZMH(JM1) - ZMH(JM22-J);
    END;
    PUT SKIP LIST('   J      XMH        XPH        ZMH          ZPH');
    DO J=1 TO J2;
    ZPH(J) = ZMH(J+1);
    PUT SKIP EDIT(J, XMH(J), XPH(J), ZMH(J), ZPH(J))
       (F(4),(4)F(10,1));
    END;

    END INITAL;
```

```
                                    PAGE  6
PRNTAL;  PROC;

     DCL (M,MOMX,MOMZ,     U,V,W        )(20,40) FLOAT BIN EXT;
     DCL (ZMH(81), ZPH(80)) FLOAT BIN EXT;
     DCL (EPS, G, TWO, ZO                         ) FLOAT BIN EXT;
     DCL (IMAX,IPL,ISL,JMAX,N                     ) FIXED BIN(31) EXT;
      DCL (I,J,I1,I2) FIXED BIN,      (MIN) BUILTIN;
     DCL (MT, MIJ, MOMXT, MOMZT, ET, ET1, ET2) INIT(0) FLOAT BIN;
     DCL  SMJ(IMAX) FLOAT BIN;
        SMJ=ZO;

     I1 = MIN(10,IMAX);
     I2 = MIN(IMAX,20);
     PUT PAGE EDIT('FOR N = ',N)(A,F(4));
     PUT EDIT(',       *_LOOP TOTAL = ',ISL,',          +_LOOP TOTAL = ',IPL)
         (A,F(8));
     DO I=1 TO IMAX;
     DO J=1 TO JMAX;
     MIJ = M(I,J);
     IF MIJ<EPS THEN U(I,J), W(I,J) = ZO;
              ELSE DO;
            U(I,J) = MOMX(I,J)/MIJ;
            W(I,J) = MOMZ(I,J)/MIJ;
                  END;
     MT = MT + MIJ;
     MOMXT = MOMXT + MOMX(I,J);
     MOMZT = MOMZT + MOMZ(I,J);

     ET1 = ET1 + MIJ * G * (ZPH(J)+ZMH(J))/TWO;
     ET2 = ET2 + MIJ * (U(I,J)**2 + W(I,J)**2)/TWO;
     END;    END;

     ET = ET1 + ET2;

NEQZ:
     DO I=1 TO IMAX;
     DO J=1 TO JMAX;
     SMJ(I) = SMJ(I) + M(I,J);
     END;    END;

     PUT SKIP(2) LIST('         M      ');
     PUT DATA(MT);
     DO J=JMAX TO 1 BY -1;
     PUT SKIP EDIT(J,(M(I,J) DO I=1 TO I1))(F(2),10 F(12,5));
     END;
```

```
                       PAGE  7
IF IMAX>10 THEN DO;                                                PRNT 460
PUT SKIP;                                                          PRNT 470
DO J=JMAX TO 1 BY -1;                                              PRNT 480
PUT SKIP EDIT(J,(M(I,J) DO I=11 TO I2))(F(2),10 F(12,5)));         PRNT 490
END;                                                               PRNT 500
               END;                                                PRNT 510
PUT SKIP(2) EDIT( SMJ )(X(2),10 F(12,5));                          PRNT 520
PUT SKIP(2) LIST('      MOMX   ');                                 PRNT 530
PUT DATA(MOMXT);                                                   PRNT 540
DO J=JMAX TO 1 BY -1;                                              PRNT 550
PUT SKIP EDIT(J,(MOMX(I,J) UO I=1 TU I1))(F(2),10 F(12,5)));       PRNT 560
END;                                                               PRNT 570
IF IMAX>10 THEN DO;                                                PRNT 580
PUT SKIP;                                                          PRNT 590
DO J=JMAX TO 1 BY -1;                                              PRNT 600
PUT SKIP EDIT(J,(MOMX I,J) UO I=11 TO I2))(F(2),10 F(12,5)));      PRNT 610
END;                                                               PRNT 620
               END;                                                PRNT 630
PUT SKIP(2) LIST('      MOMZ   ');                                 PRNT 640
PUT DATA(MOMZT);                                                   PRNT 650
DO J=JMAX TO 1 BY -1;                                              PRNT 660
PUT SKIP EDIT(J,(MOMZ(I,J) UO I=1 TU I1))(F(2),10 F(12,5)));       PRNT 670
END;                                                               PRNT 680
IF IMAX>10 THEN DO;                                                PRNT 690
PUT SKIP;                                                          PRNT 700
DO J=JMAX TO 1 BY -1;                                              PRNT 710
PUT SKIP EDIT(J,(MOMZ(I,J) UU I=11 TO I2))(F(2),10 F(12,5)));      PRNT 720
END;                                                               PRNT 730
               END;                                                PRNT 740
PUT SKIP(2) DATA(FT1, ET2, ET);                                    PRNT 750
IF N=0 THEN RETURN;                                                PRNT 760
PUT SKIP(2) LIST('      U');                                       PRNT 770
DO J=JMAX TO 1 BY -1;                                              PRNT 780
PUT SKIP EDIT(J,(U(I,J) DO I=1 TO I1))(F(2),10 F(12,5)));          PRNT 790
END;                                                               PRNT 800
IF IMAX>10 THEN DO;                                                PRNT 810
PUT SKIP;                                                          PRNT 820
DO J=JMAX TO 1 BY -1;                                              PRNT 830
PUT SKIP EDIT(J,(U(I,J) DO I=11 TO I2))(F(2),10 F(12,5)));         PRNT 840
END;                                                               PRNT 850
               END;                                                PRNT 860
PUT SKIP(2) LIST('      W');                                       PRNT 870
DO J=JMAX TO 1 BY -1;                                              PRNT 880
PUT SKIP EDIT(J,(W(I,J) DO I=1 TO I1))(F(2),10 F(12,5)));          PRNT 890
END;                                                               PRNT 900
```



```
                         PAGE  8
IF IMAX>10 THEN DO;                                                PRNT 910
PUT SKIP;                                                          PRNT 920
DO J=JMAX TO 1 BY -1;                                              PRNT 930
PUT SKIP EDIT(J,(W(I,J) DO I=11 TO I2))(F(2),10 F(12,5));          PRNT 940
END;                                                               PRNT 950
                END;                                               PRNT 960
PUT SKIP(2) LIST('       V');                                      PRNT 970
DO J=JMAX TO 1 BY -1;                                              PRNT 980
PUT SKIP EDIT(J,(V(I,J) DO I=1 TO I1))(F(2),10 F(12,5));           PRNT 990
END;                                                               PRNT1000
IF IMAX>10 THEN DO;                                                PRNT1010
PUT SKIP;                                                          PRNT1020
DO J=JMAX TO 1 BY -1;                                              PRNT1030
PUT SKIP EDIT(J,(V(I,J) DO I=11 TO I2))(F(2),10 F(12,5));          PRNT1040
END;                                                               PRNT1050
                END;                                               PRNT1060
                                                                   PRNT1070
END PRNTAL;                                                        PRNT1080
```

```
                                PAGE  9
PSAQS:  PROC;
    DCL (ERFC, EXP ) BUILTIN;
    DCL (Q,Q1) (40,40) FLOAT BIN EXT;
    DCL (P,P1          )(20,40) FLOAT BIN EXT;
    DCL (DX(20), DZ(40), XMH(81),ZMH(81), XPH(60),ZPH(80))
                                               FLOAT BIN EXT;
    DCL (IMAX, JMAX) FIXED BIN(31) EXT;
    DCL (PCON, SQDA, TPCON, TSDA, TXPH, TWO  ) FLOAT BIN EXT;
    DCL P1CON FLOAT BIN;
    DCL (I, J, K, L) FIXED BIN(31);

    PUT PAGE LIST('     THE P(I,J)"S');
    TXPH = 2.0 * XPH( IMAX );

    DO I=1 TO IMAX;
       DO K=1 TO IMAX;
       IF I¬=K THEN DO;
       P(I,K) = PFUN( (XMH(K)+XMH(I))/TSDA )
              + PFUN( (XPH(K)+XPH(I))/TSDA )
    + PFUN( (TXPH - XPH(K) - XPH(I))/TSDA )
    - PFUN( (TXPH - XMH(K) - XPH(I))/TSDA )
    - PFUN( (TXPH - XPH(K) - XMH(I))/TSDA )
    + PFUN( (TXPH - XMH(K) - XMH(I))/TSDA )
    -  PFUN( (XPH(K)+XMH(I))/TSDA ) - PFUN( (XMH(K)+XPH(I))/TSDA ) ;

       IF K>I THEN  P(I,K) = P(I,K) +
       PFUN( (XMH(K)-XPH(I))/TSDA ) - PFUN( (XPH(K)-XPH(I))/TSDA )
    -  PFUN( (XMH(K)-XMH(I))/TSDA ) + PFUN( (XPH(K)-XMH(I))/TSDA ) ;

              ELSE  P(I,K) = P(I,K) +
       PFUN( (XMH(I)-XPH(K))/TSDA ) - PFUN( (XPH(I)-XPH(K))/TSDA )
    -  PFUN( (XMH(I)-XMH(K))/TSDA ) + PFUN( (XPH(I)-XMH(K))/TSDA ) ;

                     END;
       END;
    END;

    DO I=1 TO IMAX;
    P(I,I)=DX(I);
       DO K=1 TO IMAX;
       IF K¬=I THEN P(I,I) = P(I,I) - P(I,K);
       END;
    END;
```

```
                                   PAGE  10
DO I=1 TO IMAX;
PUT SKIP(2) EDIT( I )(F(5) );
PUT EDIT( (P(I,J) DO J=1 TO IMAX) )(F(12,5));
END;

PUT SKIP(4) LIST('     THE Q(I,J)"S');
DO J=1 TO JMAX;
   DO L=1 TO JMAX;
   IF J=L THEN Q(J,J)= DZ(J) - 2*PCON +
         2.0*PFUN( DZ(J)/TSDA )
   + PFUN( ZMH(J)/SQDA ) - 2.0*PFUN( (ZMH(J)+ZPH(J))/TSDA )
   + PFUN( ZPH(J)/SQDA );

         ELSE DO;
   Q(J,L) = PFUN( (ZMH(L)+ZMH(J))/TSDA ) -
   PFUN( (ZPH(L)+ZMH(J))/TSDA )- PFUN( (ZMH(L)+ZPH(J))/TSDA )
   + PFUN( (ZPH(L)+ZPH(J))/TSDA );

   IF J<L THEN Q(J,L) = Q(J,L) + PFUN( (ZMH(L)-ZPH(J))/TSDA )
- PFUN( (ZPH(L)-ZPH(J))/TSDA ) - PFUN( (ZMH(L)-ZMH(J))/TSDA )
   + PFUN( (ZPH(L)-ZMH(J))/TSDA );

         ELSE Q(J,L) = Q(J,L) +PFUN((ZMH(J)-ZPH(L))/TSDA)
  - PFUN( (ZPH(J)-ZPH(L))/TSDA )- PFUN( (ZMH(J)-ZMH(L))/TSDA )
         + PFUN( (ZPH(J)-ZMH(L))/TSDA );
         END;
   END;
END;
DO I=1 TO JMAX;
PUT SKIP(2) EDIT( I )(F(5) );
PUT EDIT( (Q(I,J) DO J=1 TO JMAX) )(F(12,5));
END;

PUT PAGE LIST('     THE P1(I,J)"S');
DO I=1 TO IMAX;
P1CON = DX(I) - TPCON;

P1(I,I) = TWO*PFUN( DX(I)/TSDA ) + P1CON
   - PFUN( XMH(I)/SQDA ) - PFUN( XPH(I)/SQDA )
   + TWO*PFUN( (XMH(I)+XPH(I))/TSDA )
   - PFUN( (XPH(IMAX)-XPH(I))/SQDA )
   + TWO*PFUN( (TXPH - XPH(I)-XMH(I))/TSDA )
   - PFUN( (XPH(IMAX) - XMH(I))/SQDA );
```

```
                              PAGE  11
   DO K=1 TO IMAX;
   IF I¬=K THEN DO;
   IF K>I THEN P1CON = PFUN( (XMH(K)-XPH(I))/TSDA )
 - PFUN( (XPH(K)-XPH(I))/TSDA ) - PFUN( (XMH(K)-XMH(I))/TSDA )
 + PFUN( (XPH(K)-XMH(I))/TSDA );
         ELSE P1CON = PFUN( (XMH(I)-XPH(K))/TSDA )
 - PFUN( (XPH(I)-XPH(K))/TSDA ) - PFUN( (XMH(I)-XMH(K))/TSDA )
 + PFUN( (XPH(I)-XMH(K))/TSDA );

   P1(I,K) = TWO * P1CON - P(I,K);
                  END;
   END;
END;
DO I=1 TO IMAX;
PUT SKIP(2) EDIT( I )(F(5) );
PUT EDIT( (P1(I,J) DO J=1 TO IMAX) )(F(12,5));
END;
PUT SKIP(4) LIST('     THE Q1(I,J)"S');
DO J=1 TO JMAX;
Q1(J,J) = DZ(J) - TPCON + TWO*PFUN( DZ(J)/TSDA )
   - PFUN( ZMH(J)/SQDA ) - PFUN( ZPH(J)/SQDA )
   + TWO * PFUN( (ZMH(J)+ZPH(J))/TSDA );
   DO L=1 TO JMAX;
IF J¬=L THEN Q1(J,L) = Q(J,L) - TWO*PFUN((ZMH(J)+ZMH(L))/TSDA)
+ TWO*PFUN( (ZMH(J)+ZPH(L))/TSDA )
+ TWO*PFUN( (ZPH(J)+ZMH(L))/TSDA )
- TWO*PFUN( (ZPH(J)+ZPH(L))/TSDA );
   END;
END;
DO I=1 TO JMAX;
PUT SKIP(2) EDIT( I )(F(5) );
PUT EDIT( (Q1(I,J) DO J=1 TO JMAX) )(F(12,5));
END;

DCL DA FLOAT BIN EXT;
DCL (DXM,DZM)(41) FLOAT BIN;
DCL (X, CP, CZ, CM)(IMAX) FLOAT BIN,
     (Z, EP, EM, EZ)(JMAX) FLOAT BIN,
(DX2, DZ2, DM1, DM2, D1, D3, CZM, CZP, EZM, TDA) FLOAT BIN;
DCL I1 FIXED BIN(31);

DO I=1 TO IMAX;
X(I) = (XPH(I)+XMH(I))/TWO;
END;
DO I=2 TO IMAX;
DXM(I) = X(I) - X(I-1);
END;
```

```
                          PAGE  12
   DO J=1 TO JMAX;
   Z(J) = (ZPH(J)+ZMH(J))/TWO;
   END;
DO J=2 TO JMAX;
   DZM(J) = Z(J) - Z(J-1);
   END;


DXM(1) = DX(1);        DZM(1) = DZ(1);
DXM(IMAX+1) = DX(IMAX);           DZM(JMAX+1) = DZ(JMAX);

TDA = TWO*DA;
DO I=1 TO IMAX;
DX2 = DX(I)**2 / 12.0;
DM1 = DXM(I)**2;
DM2 = DXM(I+1)**2;
IF I=1 THEN D1 = DX(1)**2;
       ELSE D1 = DX(I-1)**2;
IF I=IMAX THEN D3 = DX(IMAX)**2;
          ELSE D3 = DX(I+1)**2;
D1 = D1/12.0;          D3 = D3/12.0;
CP(I) = TDA/(DM2+D3-DX2+DXM(I)*DXM(I+1)+(D1-DX2)*DAM(I+1)/DXM(I));
CM(I) = TDA/(DM1+D1-DX2+DXM(I)*DXM(I+1)+(D3-DX2)*DAM(I)/DXM(I+1));
CZ(I) = 1.0E0 - CP(I) - CM(I);
END;

   DO J=1 TO JMAX;
   DZ2 = DZ(J)**2/12.0;
DM1 = DZM(J)**2;
DM2 = DZM(J+1)**2;
   IF J=1 THEN D1 = DZ(1)**2;
          ELSE D1 = DZ(J-1)**2;
   IF J=JMAX THEN D3 = DZ(JMAX)**2;
             ELSE D3 = DZ(J+1)**2;
   D1 = D1/12.0;          D3 = D3/12.0;
   EP(J) = TDA/(DM2+D3-DZ2+DZM(J)*DZM(J+1)+(D1-DZ2)*DZM(J+1)/
                                  DZM(J));
   EM(J) = TDA/(DM1+D1-DZ2+DZM(J)*DZM(J+1)+(D3-DZ2)*DZM(J)/
                                  DZM(J+1));
   EZ(J) = 1.0E0 - EP(J) - EM(J);
   END;
```

```
                              PAGE  13

CZM = CZ(1)+CM(1);
IF CZM >= 0. THEN DO;
   P(1,*), P1(1,*) = 0.;
   P(1,1) = CZM * DX(1);
   P(1,2) = CP(1) * DX(1);
   P1(1,1) = (CZ(1)-CM(1))* DX(1);                    P1(1,2) = P(1,2);
         END;

DO I=2 TO IMAX-1;
IF CZ(I) >= 0. THEN DO;
   P(I,*), P1(I,*) = 0.;
   P(I,I-1), P1(I,I-1) = CM(I)*DX(I);
   P(I,I), P1(I,I) = CZ(I)*DX(I);
   P(I,I+1), P1(I,I+1) = CP(I)*DX(I);
         END;
END;

CZP=CZ(IMAX) + CP(IMAX);
IF CZP >= 0. THEN DO;
   I1=IMAX-1;       P(IMAX,*), P1(IMAX,*) = 0.;
   P(IMAX,I1), P1(IMAX,I1) = CM(IMAX) * DX(IMAX);
   P(IMAX,IMAX) = CZP * DX(IMAX);
  P1(IMAX,IMAX) = (CZ(IMAX)-CP(IMAX))*DX(IMAX);
               END;


PUT PAGE LIST('NEW THE P(I,J)"S');
DO I=1 TO IMAX;
PUT SKIP(2) EDIT( I )(F(5) );
PUT EDIT( (P(I,J) DO J=1 TO IMAX) )(F(12,5));
END;
PUT PAGE LIST('NEW THE P1(I,J)"S');
DO I=1 TO IMAX;
PUT SKIP(2) EDIT( I )(F(5) );
PUT EDIT( (P1(I,J) DO J=1 TO IMAX) )(F(12,5));
END;

   EZM = EZ(1)+EM(1);
   IF EZM >= 0. THEN DO;
        Q(1,*), Q1(1,*) = 0.;
        Q(1,1) = EZM * DZ(1);
        Q1(1,1) = (EZ(1)-EM(1)) * DZ(1);
        Q(1,2), Q1(1,2) = EP(1)*DZ(1);
                END;
```

```
                                   PAGE  14
            DO J=2 TO JMAX-1;                                                  PSAQ2210
            IF EZ(J) >= 0 THEN DO;                                             PSAQ2220
                 Q(J,*), Q1(J,*) = 0.;                                         PSAQ2230
                 Q(J,J-1), Q1(J,J-1) = EM(J)*DZ(J);                            PSAQ2240
                 Q(J,J), Q1(J,J) = EZ(J) * DZ(J);                              PSAQ2250
                 Q(J,J+1), Q1(J,J+1) = EP(J) * DZ(J);                          PSAQ2260
                                   END;                                        PSAQ2270
            END;                                                               PSAQ2280
                                                                               PSAQ2290
        IF EZ(JMAX) >= 0 THEN DO;                                              PSAQ2300
                 Q(JMAX,*), Q1(JMAX,*) = 0;                                    PSAQ2310
                 Q(JMAX,JMAX-1), Q1(JMAX,JMAX-1) = EM(JMAX) * DZ(JMAX);        PSAQ2320
                 Q(JMAX,JMAX), Q1(JMAX,JMAX) = EZ(JMAX) * DZ(JMAX);            PSAQ2330
                               END;                                            PSAQ2340
                                                                               PSAQ2350
                                                                               PSAQ2360
        PUT SKIP(4) LIST('NEW THE Q(I,J)''S');                                 PSAQ2370
        DO I=1 TO JMAX;                                                        PSAQ2380
        PUT SKIP(2) EDIT( I )(F(5) );                                          PSAQ2390
        PUT EDIT( (Q(I,J) DO J=1 TO JMAX) )(F(12,5));                          PSAQ2400
        END;                                                                   PSAQ2410
        PUT SKIP(4) LIST('NEW THE Q1(I,J)''S');                                PSAQ2420
        DO I=1 TO JMAX;                                                        PSAQ2430
        PUT SKIP(2) EDIT( I )(F(5) );                                          PSAQ2440
        PUT EDIT( (Q1(I,J) DO J=1 TO JMAX) )(F(12,5));                         PSAQ2450
        END;                                                                   PSAQ2460
                                                                               PSAQ2470
                                                                               PSAQ2480
PFUN;   PROC ( Z ) RETURNS(FLOAT BIN);                                         PSAQ2490
        DCL (Z,PZ) FLOAT BIN;                                                  PSAQ2500
        PZ = PCON * EXP(-(Z**2)) - Z*SQDA * ERFC(Z);                           PSAQ2510
        RETURN( PZ );                                                          PSAQ2520
        END PFUN;                                                              PSAQ2530
                                                                               PSAQ2540
/*      --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- */PSAQ2550
                                                                               PSAQ2560
         END PSAQS;                                                            PSAQ2570
```

```
                              PAGE  15
MASMON:  PROC;

    DCL (M,MOMX,MOMZ,RHO,U,  W        )(20,40) FLOAT BIN EXT;
    DCL (TMOMX, TMOMZ, TM) (40,80) FLOAT BIN EXT;
    DCL (DX(20), DZ(40), XMH(81),ZMH(81), XPH(60),ZPH(80))
                                          FLOAT BIN EXT;
    DCL (ETAM(2), ETAP(2), XIM(2), XIP(2))    FLOAT BIN;
    DCL (DT, EPS, G                          ) FLOAT BIN EXT;
    DCL (IMAX, JMAX, I2, J2    ) FIXED BIN(31) EXT;
    DCL ERR BIT(1) EXT;

    DCL (FLOOR, MAX, MIN) BUILTIN;
    DCL(A1,A2,GDT,TEMP,TMAX,UDT,XM,XMAX,XP,ZM,ZMAX,ZP)FLOAT BIN;
    DCL(I,IP,I1,I3,J,J3,K,KM,L,LM,LP,NP,N1) FIXED BIN(31);

    GDT = G*DT;
    MOMZ = MOMZ - GDT * M;

    DO I=1 TO IMAX;
    DO J=1 TO JMAX;
    IF M(I,J)<EPS THEN U(I,J),w(I,J)=0;
              ELSE DO;
          U(I,J) = MOMX(I,J) / M(I,J);
          W(I,J) = MOMZ(I,J) / M(I,J);
                 END;
    RHO(I,J) = M(I,J) / (DX(I)*DZ(J));
    END;    END;

    TM, TMOMX, TMOMZ = 0;
LOOPI:  DO I=1 TO IMAX;
        DO J=1 TO JMAX;
/******  SUBROUTINE 1  *****/
    ZM = ZMH(J) + w(I,J)*DT;
    ZP = ZPH(J) + w(I,J)*DT;
    NP=0;   ETAM, ETAP, XIM, XIP = 0;
    ZMAX = ZPH(JMAX);

    IF ZP >= ZMAX THEN DO;
      IF ZM < ZMAX THEN DO;
          NP=1;  ETAM(1)=ZM;   ETAP(1)=ZMAX;
                  END;
                 END;

              ELSE DO;  /*  NOW FOR ZP<ZMAX  */
      IF ZM <= -ZMAX THEN DO;
          IF ZP > -ZMAX THEN DO;
            NP=1;  ETAM(1)=ZMAX;  ETAP(1) = 2*ZMAX + ZP;
                        END;
                  END;
                  PAGE  16
                         ELSE DO;   /* NOW FOR ZM > -ZMAX   */
         IF ZP > 0 THEN DO;
            IF ZM>=0 THEN DO;
                NP=1;  ETAM(1)=ZM;  ETAP(1)=ZP;
                          END;
                        ELSE DO;
         NP=2; ETAP(1)=ZP;
         ETAM(2) = 2*ZMAX + ZM;
      ETAP(2) = 2*ZMAX;
                          END;
                        END;  /* FOR ZP>0   */

      ELSE DO;
      NP=1;     ETAM(1) = ZM + 2*ZMAX;
                ETAP(1) = ZP + 2*ZMAX;
          END;
                END;
                          END;  /*  FOR ZP<ZMAX  */

/*  --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- */

      IF NP>0 THEN DO;
/******  SUBROUTINE 2    *********/
      UDT = U(I,J) * DT;
      XMAX = XPH(IMAX);    TMAX = 2*XMAX;
      TEMP = (XMH(I) + UDT)/TMAX;
      XM = TMAX * (TEMP - FLOOR(TEMP));

      TEMP = (XPH(I)+UDT)/TMAX;
      XP = TMAX * (TEMP - FLOOR(TEMP));

      IF XP > XM THEN DO;
         IP=1;    XIM(1)=XM;    XIP(1)=XP;
                      END;
                  ELSE DO;
         IP = 2;  XIM(1) = XM;  XIP(2) = XP;   XIP(1) = TMAX;
                      END;

/********** SUBROUTINE 3    **********/

      DO I1=1 TO IP;
      DO KM=1 TO I2;
      IF XMH(KM)<=XIM(I1) THEN IF XIM(I1)<=XPH(KM) THEN GO TO NEXTI1;
      END;
        PUT SKIP LIST('ERROR-1 NO KM');   GO TO ERROUT;
```

```
                              PAGE  17
NEXT1;  DO KP=KM TO I2;
        IF XMH(KP)<XIP(I1) THEN IF XIP(I1)<=XPH(KP) THEN GU TO NEXT2;
        END;
     PUT SKIP LIST('ERROR-2 NO KP');    GU TU ERROUT;
NEXT2;  DO N1=1 TO NP;
          DO LM=1 TO J2;
      IF ZMH(LM)<=ETAM(N1) THEN IF ETAM(N1)<=ZPH(LM) THEN GU TO NEXT3;
          END;
     PUT SKIP(2) DATA(I,J,NP,IP,I2,N1,ETAM,ETAP,ZMH,ZPH);
     PUT SKIP LIST('ERROR-3 NO LM');    GU TO ERROUT;
NEXT3:    DO LP=LM TO J2;
        IF ZMH(LP)<ETAP(N1) THEN IF ETAP(N1)<=ZPH(LP) THEN GU TO NEXT4;
      END;
     PUT SKIP LIST('ERROR-4 NO LP');    GU TU ERROUT;

NEXT4;

            DO K=KM TO KP;
            DO L=LM TO LP;
     A1 = MIN(XPH(K),XIP(I1)) - MAX(XMH(K),XIM(I1));
     A2 = MIN(ZPH(L),ETAP(N1)) - MAX(ZMH(L),ETAM(N1));
      .
     TM(K,L) = TM(K,L) + RHO(I,J) * A1 * A2;
     TMOMX(K,L) = TMO4X(K,L) + RHO(I,J)*U(I,J) * A1 * A2;
     TMOMZ(K,L) = TMOMZ(K,L) + RHO(I,J)*W(I,J) * A1 * A2;
            END;    ENU;   /*  ENU OF K AND L LOOPS      */
     END;   END;   /*  END OF I1 AND N1 LOOPS  */

                END;          /* END OF NP>0 DO    */
       END;  /*  FOR J LOOP    */
     ENU LOOPI;    /*  END OF I LUOP      */

     I3=I2+1;    J3=J?+1;

     DO I=1 TO IMAX;
     DO J=1 TO JMAX;
     M(I,J) = TM(I,J) + TM(I3-I,J) + TM(I,J3-J) + TM(I3-I,J3-J);
     MOMX(I,J)=TMOMX(I,J)-TMUMX(I3-I,J)+TMOMX(I,J3-J)-TMOMA(I3-I,J3-J);
     MOMZ(I,J)=TMOM7(I,J)+TMUMZ(I3-I,J)-TMOMZ(I,J3-J)-TMUMZ(I3-I,J3-J);
     END;   END;

     RETURN;
ERROUT;   ERR='1'B;
     END MASMON;
```

```
                                   PAGE  18
DENSTY:  PROC;
      DCL (DEBUG, ERR) BIT(1) EXT;

      DCL Q(40,40) FLOAT BIN EXT;
      DCL (M, M2VX, M2VZ, P,     RHO, V, V1)        (20,40) FLOAT BIN EXT;
      DCL (DX(20), DZ(40), XMH(81),ZMH(81), XPH(60),ZPH(80))
                                                FLOAT BIN EXT;
      DCL (DA, DG, EPS1, RHOC, VMAX, ZO, ONE, TWO) FLOAT BIN EXT;

      DCL (MAX, MIN) BUILTIN;
      DCL (IMAX, JMAX, ISL) FIXED BIN(31) EXT;
      DCL (I, ICST1, J, K, L) FIXED BIN(31);
      DCL (PRHO, PSUM1,PSUM2, QRHO,QSUM1,QSUM2,RHOMAX,XMPI,ZMPJ
               ) FLOAT BIN;

      V, M2VX, M2VZ = 0;
      ICST1 = 0;
      VMAX = ZO;
 /*   --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- */
STAR_LOOP:
      IF DEBUG THEN PUT SKIP LIST('ENTERED STAR_LOOP');
      ISL = ISL + 1;
      ICST1 = ICST1 + 1;;
      DO I=1 TO IMAX;
      DO J=1 TO JMAX;
      RHO(I,J) = M(I,J) / (DX(I)*DZ(J));
      IF I+J=2 THEN RHOMAX = RHO(1,1)-RHOC;
               ELSE RHOMAX = MAX(RHOMAX,(RHO(I,J)-RHOC));
      END;   END;

      IF DEBUG THEN PUT SKIP DATA(ICST1, RHOMAX);
      IF RHOMAX <=EPS1 THEN GO TO FINAL_MOMXAZ;

      VMAX = ZO;
      DO I=1 TO IMAX;
      XMPI =  XMH(I) + XPH(I);
      DO J=1 TO JMAX;
      PSUM1, PSUM2 = ZO;
         DO K=1 TO IMAX;
         PRHO = P(K,I) * MAX(RHO(K,J)-RHOC, ZO);
         PSUM1 = PSUM1 + PRHO;
         PSUM2 = PSUM2 + PRHO*(XMPI - XMH(K) - XPH(K)) / TWO;
         END;
      V(I,J) = V(I,J) + DA * MAX(RHO(I,J)-RHOC, ZO);
      VMAX = MAX(V(I,J), VMAX);
      M(I,J) = DZ(J)*(DX(I)*MIN(RHO(I,J),RHOC) + PSUM1);
      M2VX(I,J) = M2VX(I,J) + DZ(J)*PSUM2;
      END;   END;
```

```
                              PAGE  19

      DO I=1 TO IMAX;
      DO J=1 TO JMAX;
      RHO(I,J) = M(I,J) / (DX(I)*DZ(J));
      END;   END;

      DO J=1 TO JMAX;
      ZMPJ =  ZMH(J) + ZPH(J);
      DO I=1 TO IMAX;
         QSUM1, QSUM2 = ZO;
         DO L=1 TO JMAX;
         QRHO = Q(L,J)*MAX(RHO(I,L)-RHOC, ZO);
         QSUM1 = QSUM1 + QRHO;
         QSUM2 = QSUM2 + QRHO*(ZMPJ-ZMH(L)-ZPH(L))/TWO;
         END;

      M(I,J) = DX(I)*(DZ(J)*MIN(RHO(I,J), RHOC) + QSUM1);
      M2VZ(I,J) = M2VZ(I,J) + DX(I)*QSUM2;
         END;   END;

      IF ICST1 > 999 THEN DO;
      PUT PAGE LIST('ICST1 > 999  "STOP"     ');
      ERR='1'B;   RETURN;
               END;
      GO TO STAR_LOOP;

 FINAL_MOMXAZ:
      IF DEBUG THEN PUT SKIP LIST('ENTERED  FINAL_MOMXAZ');
      IF DEBUG THEN PUT SKIP DATA(VMAX);
      END DENSTY;
```

```
                              PAGE  20
MOMEN:    PROC;

      DCL (EXP )BUILTIN;
      DCL (DEBUG, ERR) BIT(1) EXT;
      DCL (Q,Q1)  (40,40)  FLOAT BIN EXT;
      DCL (P,P1,        MOMX,MOMZ,      U,V,W, V1  )(20,40) FLOAT BIN EXT;
      DCL (M2VX, M2VZ)(20,40) FLOAT BIN EXT;
      DCL (TMOMX, TMOMZ     ) (40,80) FLOAT BIN EXT;
      DCL (DX(20), DZ(40), DA, DG, DT, GMAX, RHOC, VMAX, ZO, ONE)
                                           FLOAT BIN EXT;
      DCL (IMAX, JMAX, IPL        ) FIXED BIN(31) EXT;

      DCL (A,EPDG,ETEMP,PSUM1,QSUM1,VMAX11   ) FLOAT BIN;
      DCL (I, IP, J, K ,L) FIXED BIN(31);

      VMAX11 = 1.1 * VMAX;
      IP=0;
      A = 1.1 * VMAX/RHOC;
      DG = DA/A;
      IF DEBUG THEN PUT SKIP DATA(VMAX11, A, DG, GMAX);

      DO I=1 TO IMAX;
      DO J=1 TO JMAX;
      V1(I,J) = V(I,J) / VMAX11;
      TMOMX(I,J) = MOMX(I,J) + M2VX(I,J) / DT;
      TMOMZ(I,J) = MOMZ(I,J) + M2VZ(I,J) / DT;
      MOMX(I,J), MOMZ(I,J) = ZO;
      END;   END;
      IF DEBUG THEN DO;
      PUT SKIP(2) LIST('      TMOMX');
      DO J=10 TO 1 BY -1;
      PUT SKIP EDIT(J,(TMOMX(I,J) DO I=1 TO 10))(F(2),10 E(12,4));
      END;
      PUT SKIP(2) LIST('      TMOMZ');
      DO J=10 TO 1 BY -1;
      PUT SKIP EDIT(J,(TMOMZ(I,J) DO I=1 TO 10))(F(2),10 E(12,4));
      END;
                     END;

PLUS_LOOP:
      IF DEBUG THEN PUT SKIP LIST('ENTERED  PLUS_LOOP');
      IF DEBUG THEN PUT SKIP DATA(IP);
      IPL = IPL + 1;
      IF (IP+1)*DG > GMAX THEN DO;
        EPDG = EXP(-IP*DG);
        DO I=1 TO IMAX;
        DO J=1 TO JMAX;
        MOMX(I,J) = MOMX(I,J) + TMOMX(I,J)*EPDG;
        MOMZ(I,J) = MOMZ(I,J) + TMOMZ(I,J)*EPDG;
        END;  END;
        RETURN;
```

```
                    PAGE  21
                                        END;

ETEMP = EXP(-IP*DG)*(ONE - EXP(-DG));
   DO I=1 TO IMAX;
   DO J=1 TO JMAX;
   MOMZ(I,J) = MOMZ(I,J) + TMOMZ(I,J)*ETEMP;
   MOMX(I,J) = MOMX(I,J) + TMOMX(I,J)*ETEMP;
   END;    END;

U, W = ZO;
DO I=1 TO IMAX;
DO J=1 TO JMAX;
   DO L=1 TO JMAX;
   U(I,J) = U(I,J) + Q(L,J)*V1(I,L)*TMOMX(I,L)/(DX(I)*DZ(L));
   W(I,J) = W(I,J) + Q1(L,J)*V1(I,L)*TMOMZ(I,L)/(DX(I)*DZ(L));
   END;
END;    END;

DO I=1 TO IMAX;
DO J=1 TO JMAX;
QSUM1, PSUM1 = ZO;
   DO K=1 TO IMAX;
PSUM1 = PSUM1 + P1(K,I)*U(K,J);
QSUM1 = QSUM1 + P(K,I)*W(K,J);
   END;

TMOMX(I,J) = (ONE-V1(I,J)) * TMOMX(I,J) + PSUM1;

TMOMZ(I,J) = (ONE-V1(I,J)) * TMOMZ(I,J) + QSUM1;
END;    END;

IP = IP + 1;
GO TO PLUS_LOOP;
END MOMEN;
```

GLOSSARY

c_i^-	Coefficient in representation of $S_x(\cdot)$ by finite difference operator
c_i^0	Coefficient in representation of $S_x(\cdot)$ by finite difference operator
c_i^+	Coefficient in representation of $S_x(\cdot)$ by finite difference operator
D	Flow domain, domain in Eq. 33
D_c	Computational domain
e_j^-	Coefficient in representation of $S_z(\cdot)$ by finite difference operator
e_j^0	Coefficient in representation of $S_z(\cdot)$ by finite difference operator
e_j^+	Coefficient in representation of $S_z(\cdot)$ by finite difference operator
$f(\cdot)$	Function defined in Eq. 10b.
g	Gravitational constant
i	Non-negative integer
j	Non-negative integer
I	Number of computational cells in horizontal direction
J	Number of computational cells in vertical direction
k	Non-negative integer
ℓ	Non-negative integer
m_{ij}	Mass in R_{ij}
$\tilde{m}_{ij}$	Intermediate mass in R_{ij}

$\bar{m}_{ij}$	Mass in cell R_{ij} at end of time step
n	Non-negative integer, unit vector normal to boundary
$P(\cdot)$	Function defined by Eq. 73b
P_{ik}	Approximation to operator for horizontal diffusion with reflection at $x = 0$ and $x = X$
$\tilde{P}_{ik}$	Approximation to operator for horizontal diffusion in infinite space
$Q_{j\ell}$	Approximation to operator for vertical diffusion with reflection at $z = 0$
$\tilde{Q}_{j\ell}$	Approximation to operator for vertical diffusion in infinite space
R_{ij}	Rectangle defined by Eq. 36
$S(\cdot)$	Operator that transforms solution at one time into solution at later time
$\bar{S}(\cdot)$	Approximation to operator that updates solution by one time step
$S_x(\cdot)$	Operator for diffusion in horizontal direction in infinite space
$S_z(\cdot)$	Operator for diffusion in vertical direction in infinite space
t	Time
u	Velocity components, horizontal velocity component
u^0	Initial velocity
$\tilde{u}$	Intermediate velocity
$\bar{u}$	Velocity at end of time step
u_{ij}	Horizontal component of velocity in R_{ij}
v	Quantity defined in Eq. 12

v_{ij}	Approximate value of v in cell R_{ij}
w	Vertical velocity component
w_{ij}	Vertical component of velocity in R_{ij}
x	Spatial coordinates, horizontal spatial coordinate
X	Width of computational domain
x_i	Horizontal coordinate of centers of computational cells in ith column
$x_{i+1/2}$	Horizontal coordinate of right-hand sides of computational cells in ith column
Δx_i	Widths of computational cells in ith column
z	Vertical spatial coordinate
Z	Height of computational domain
z_j	Vertical coordinate of centers of computational cells in jth row
$z_{j+1/2}$	Vertical coordinate of tops of computational cells in jth row
Δz_j	Heights of computational cells in jth row
α	Pseudo-time variable
$\Delta\alpha$	Step of pseudo-time variable α
γ	Pseudo-time variable
γ_0	Cut-off parameter for solution of Eq. 13a
$\Delta\gamma$	Step of pseudo-time variable γ
Δ	Laplacian operator
ε	Cut-off to keep from dividing by zero in Eq. 38
ε_1	Cut-off in Eq. 81 for approximate satisfaction of density constraint

ε_2	**Cut-off parameter that determines how Eq. 13a is solved**
μ_{xij}	**Horizontal component of momentum in R_{ij}**
$\tilde{\mu}_{xij}$	**Intermediate horizontal component of momentum in R_{ij}**
μ_{zij}	**Vertical component of momentum in R_{ij}**
$\tilde{\mu}_{zij}$	**Intermediate vertical component of momentum in R_{ij}**
$\bar{\mu}_{ij}$	**Momentum in cell R_{ij} at end of time step**
ρ	**Density**
ρ_0	**Density of liquid phase**
ρ^0	**Initial density**
$\tilde{\rho}$	**Intermediate density**
$\bar{\rho}$	**Density at end of time step**
τ	**Time step**
∇	**Gradient**

INITIAL DISTRIBUTION EXTERNAL TO THE APPLIED PHYSICS LABORATORY*

The work reported in TG 1325 was done under Navy Contract N00024-78-C-5384. This work is related to Task ZL50, which is supported by Office of Naval Research (Code 438).

ORGANIZATION	LOCATION	ATTENTION	No. of Copies
DEPARTMENT OF DEFENSE			
DDC	Alexandria, VA 22314		12
Department of the Navy			
NAVPRO	Laurel, MD 20810		1
NAVSEASYSCOM	Washington, DC 20362	SEA-032	1
		SEA-0322	1
		SEA-03B	1
		SEA-03512	1
		SEA-09G3	2
DTNSRDC, Annapolis Lab	Annapolis, MD 21402	Tech. Lib.	1
DTNSRDC	Bethesda, MD 20084	Code 01	1
		Code 15	1
		Code 18	1
		Code 1521	1
		Code 1541	1
		Code 1552	2
		Code 1843	1
		Code 5641	1
U.S. Naval Academy	Annapolis, MD 21402	B. Johnson	1
		Lib.	1
Office of Naval Res., Branch Off.	Boston, MA 02210	Dir.	1
	Chicago, IL 60605	Dir.	1
	New York, NY 10003	Dir.	1
	Pasadena, CA 91101	Dir.	1
	San Francisco, CA 94102	Dir.	1
Office of Naval Research, Scientific Liaison Group, American Embassy	APO San Francisco, CA 96503		1
Office of Naval Research	Arlington, VA 22217	Code 438	3
		Code 200	1
		Code 210	1
		Code 211	1
		Code 212	1
		Code 221	1
		Code 473	1
		Code 480	1
		Code 481	1
NAVAIRSYSCOM	Washington, DC 20361	AIR-03	1
		AIR-03B	1
		AIR-310	1
		AIR-5301	1
		AIR-50174	2
SSPO	Washington, DC 20376	Dir.	1
		SP-2022	1
Puget Sound Naval Shipyard	Bremerton, WA 98314	Commander	1
Charleston Naval Shipyard	Charleston, SC 29408	Commander	1
Long Beach Naval Shipyard	Long Beach, CA 90801	Commander	1
Philadelphia Naval Shipyard	Philadelphia, PA 19112	Tech. Lib.	1
Portsmouth Naval Shipyard	Portsmouth, NH 03801	Commander	1
Norfolk Naval Shipyard	Portsmouth, VA 23709	Commander	1
Pearl Harbor Naval Shipyard	FPO San Francisco, CA 96610	Lib.	1
Hunters Point Naval Shipyard	San Francisco, CA 94135	Tech. Lib.	1
Mare Island Naval Shipyard	Valleno, CA 94592	Tech. Lib.	1
Naval Undersea Center	San Diego, CA 92132	Code 2501	1
		Tech. Lib.	1
		Code 4007	1
Naval Research Lab.	Washington, DC 20375	Code 2627	6
		Code 4000	1
		Code 7706	1
		Code 8441	1
Naval Weapons Center	China Lake, CA 93555	Lib.	1

Requests for copies of this report from DoD activities and contractors should be directed to DDC, Cameron Station, Alexandria, Virginia 22314 using DDC Form 1 and, if necessary, DDC Form 55.

*Initial distribution of this document within the Applied Physics Laboratory has been made in accordance with a list on file in the APL Technical Publications Group.

INITIAL DISTRIBUTION EXTERNAL TO THE APPLIED PHYSICS LABORATORY*

ORGANIZATION	LOCATION	ATTENTION	No. of Copies
Department of Navy (cont'd)			
Naval Ocean Systems Ctr.	San Diego, CA 92132	Code 2501	1
		Tech. Lib.	1
		Code 5311	1
Naval Surface Weapons Center	Dahlgren, VA 22418	Tech. Lib.	1
		Comp. & Analy. Lab.	1
	Silver Spring, MD 20910	J. Enig	1
		Lab.	1
		M. Ciment	1
		A. E. Berger	1
		H. M. Glaz	1
		J. M. Solomon	1
Naval Ship Engineering Center	Philadelphia, PA 19112	Tech. Lib.	1
	Hyattsville, MD 20782	Code 6034	1
		Code 6101E	1
		Code 6110	1
		Code 6114	1
		Code 6136	1
		Code 6140	1
Naval Postgraduate School	Monterey, CA 93940	Lib.	1
Naval Underwater Systems Center	Newport, RI 02840	Tech. Lib.	1
Naval Coastal System Lab.	Panama City, FL 32401	Tech. Lib.	1
Naval Missile Center	Pt. Mugu, CA 93041	Tech. Lib.	1
Oceanographer of the Navy	Alexandria, VA 22332		1
Naval Oceanographic Office	Washington, DC 20373	Commander	1
Commandant of the Marine Corps, Scientific Advisor	Washington, DC 20380	Code AX	1
Department of the Army			
Res. & Tech. Div., Army Eng. Reactors Group	Ft. Belvoir, VA 22060		1
Army Mobility Equip. Res. Center	Ft. Belvoir, VA 22060	Tech. Doc. Center	1
Army Missile Command, Redstone Scientific Infor. Center	Redstone Arsenal, AL 35809	Chief, Doc. Sec.	1
Army Research Office	Research Triangle Pk., NC 27709		1
Office of Chief of Staff	Washington, DC 20310	Chief, Res. & Dev.	1
Department of the Air Force			
Office of Scientific Research/NA	Washington, DC 20332		1
		R. Buchal	1
AFDRD-AS/M	Washington, DC 20330		1
DEPARTMENT OF TRANSPORTATION			
Coast Guard Hq.	Washington, DC 20591	Lib. Sta. 5-2	1
U.S. GOVERNMENT AGENCIES			
National Aeronautics and Space Administration			
NASA Scientific and Tech. Info. Fac.	Baltimore/Washington International Airport		1
NASA Lewis Research Center	Cleveland, OH 44871	W. J. Anderson	1
NASA Langley Research Center	Hampton, VA 23665	D. Bushnell	1
		W. G. Pritchard	1
		M. Rose	1
		J. Sanz	1
		J. South	1
		B. Subramanian	1
		T. Zang	1
CONTRACTORS			
Ampex Corp.	Redwood City, CA 94063	A. Eshel	1
Aeronautical Res. of Princeton, Inc.	Princeton, NJ 08540	Donaldson Assoc.	1

*Initial distribution of this document within the Applied Physics Laboratory has been made in accordance with a list on file in the APL Technical Publications Group.

INITIAL DISTRIBUTION EXTERNAL TO THE APPLIED PHYSICS LABORATORY*

ORGANIZATION	LOCATION	ATTENTION	No. of Copies
CONTRACTORS (cont'd)			
Aerospace Corp.	Los Angeles, CA 90009	T. D. Taylor	1
CALSPAN Corp.	Buffalo, NY 14221	A. Ritter	1
Draper Lab.	Cambridge, MA 02139	Lib.	1
Flow Research, Inc.	Cambridge, MA 02142	S. Orszag	1
	Kent, WA 98031	D. Ko	1
		E. Murman	1
		J. Riley	1
	Los Angeles, CA 90045		1
Gulf Science & Technology	Pittsburgh, PA 15230	S. Leventhal	1
Hydronautics, Inc.	Laurel, MD 20810	J. Dunne	1
		V. Johnson	1
		M. Tulin	1
JAYCOR	Del Mar, CA 92014	R. Chan	1
		M. Vandervorst	1
		J. Young	1
Jet Propulsion Laboratory	Pasadena, CA 91125	L. Mack	1
Oceanics, Inc.	Plainview, NY 11803	P. Kaplan	1
Physical Dynamics, Inc.	La Jolla, CA 92038	E. Montroll	1
Poseidon Research	Los Angeles, CA 90049	S. Crow	1
Shaker Research Corp.	Ballston Lake, NY 12019	C. Pan	1
Southwest Research Institute	San Antonio, TX 78228	H. N. Abramson	1
		Ed., App. Mech. Rev.	1
Tetra Tech, Inc.	Pasadena, CA 91107	R. Wade	1
TRW Systems Group	Redondo Beach, CA 90278	E. Baum	1
UNIVERSITIES			
Brown University	Providence, RI 02912	J. Clarke	1
		J. Liu	1
California Institute of Technology	Pasadena, CA 91109	H. W. Liepmann	1
		M. S. Plesset	1
		A. Roshko	1
		T. Y. Wu	1
California State University	Long Beach, CA 90840	T. Cebeci	1
Carnegie Inst. of Technology	Pittsburgh, PA 15213	R. MacCamy	1
Case Western Reserve University	Cleveland, OH 44106	E. Reshotko	1
City University of New York	New York, NY 10036	H. Rauch	1
Columbia University	New York, NY 10027	V. Castelli	1
		H. Elrod	1
Cornell University	Ithaca, NY 14851	E. Resler	1
Harvard University	Cambridge, MA 02138	G. Birkhoff	1
		G. Carrier	1
Kyoto University	Kyoto, Japan	N. Takaaki	1
Lehigh University	Bethlehem, PA 18015	G. McAllister	1
Los Alamos Scientific Lab., Univ. of California	La Alamos, NM 87544	C. Hirt	1
MIT	Cambridge, MA 02139	M. Abkowitz	1
		Comm. Off. NROTC Nav. Admin. Unit	1
		L. Howard	1
		P. Mandel	1
		C. Mei	1
		E. Mollo-Christensen	1
		J. Newman	1
		R. Probstein	1
Mississippi State University	State College, MS 39762	J. Thompson	1
Northwestern University	Evanston, IL 60201	S. Davis	1
		G. Sivashinsky	1
New York University	New York, NY 10003	J. Stoker	1
Pennsylvania State University	University Park, PA 16802	J. Lumley	1
Polytechnic Inst. of New York	Farmingdale, NY 11735	M. Bloom	1
Princeton University	Princeton, NJ 08540	F. Hama	1
Purdue University	Lafayette, IN 47907	V. Goldschmidt	1
Rensselaer Polytechnic Institute	Troy, NY 12181	R. DiPrima	1
		L. Segal	1
Saint Louis University	St. Louis, MO 63103	J. Gammel	1

*Initial distribution of this document within the Applied Physics Laboratory has been made in accordance with a list on file in the APL Technical Publications Group.

INITIAL DISTRIBUTION EXTERNAL TO THE APPLIED PHYSICS LABORATORY*

ORGANIZATION	LOCATION	ATTENTION	No. of Copies
UNIVERSITIES (cont'd)			
Scripps Inst. of Oceanography, University of California	La Jolla, CA 92037	Dir.	1
Stanford University	Stanford, CA 94305	M. van Dyke	1
State University of New York	Binghamton, NY 13901	J. Geer	1
Stevens Inst. of Technology	Hoboken, NJ 07030	J. Breslin	1
		D. Savitsky	1
		T. Goodman	1
Technion-Israel Inst. of Tech.	Haifa, Israel	M. Poreh	1
University of California	Berkeley, CA 94720	Lib.	1
		P. Lieber	1
		P. Naghdi	1
		W. Webster	1
		J. Wehausen	1
	San Diego, CA 92037	W. Munk	1
	La Jolla, CA 92093	J. Miles	1
University of Chicago	Chicago, IL 60637	J. Bona	1
University of Delaware	Newark, DE 19711	J. Wu	1
University of Edinburgh	Edinburgh, Scotland	J. Martin	1
		A. Mackie	1
University of Hawaii	Honolulu, HI 96822	J. Craven	1
University of Houston	Houston, TX 77004	F. Hussain	1
University of Illinois	Urbana, IL 61803	J. Robertson	1
University of Iowa	Iowa City, IA 52242	J. Kennedy	1
		L. Landweber	1
University of Maryland	College Park, MD 20742	J. Burgers	1
		Prof. Pai	1
		A. Fuller	1
	Catonsville, MD 21228	A. Aziz	1
University of Miami	Coral Gables, FL 33124	F. Tappert	1
University of Michigan	Ann Arbor, MI 48105	T. Ogilvie	1
		C. Yih	1
		V. Phelps	1
University of Minnesota	Minneapolis, MN 55414	Straub Lib.	1
		E. Silberman	1
University of Newcastle Upon Tyne	Newcastle Upon Tyne, England	A. Jeffrey	1
University de Paris-Sud	Orsay, France	R. Temam	1
University of Southern California	Los Angeles, CA 90007	J. Laufer	1
University of Texas	Austin, TX 78712	J. Cannon	1
University of Virginia	Charlottesville, VA 22903	E. Gunter, Jr.	1
University of Washington	Seattle, WA 98105	A. Hertzberg	1
		C. Pearson	1
University of Wisconsin	Madison, WI 53706	B. Noble	1
		S. Parter	1
Virginia Polytechnic Inst. & St. Univ.	Blacksburg, VA 24060	I. Besieris	1
		J. Cochran	1
		W. Kohler	1
Virginia State College	Petersburg, VA 23803	K. Agrawal	1
MISCELLANEOUS			
Applied Institute of Mathematics, Inc.	New York, NY 10012	E. L. Reiss	2
Australian Embassy	Washington, DC 20036	Def. Res. & Dev. Attache	1
Engineering Societies Lib.	New York, NY 10017		1
Instituto de Techologia Naval	Buenos Aires, Argentina	R. A. Bastianon	1
R. H. Kraichnan	Dublin, NH 03444		1
Library of Congress, Sci. and Tech. Div.	Washington, DC 20540		1
Maritime Administration	Washington, DC 20235	Off. Res. & Dev.	1
		Div. Ship Des.	1
National Bureau of Standards	Washington, DC 20234	G. Kulin	1
		P. Klebanoff	1
		M. Ciment	1
National Research Council	Ottawa 7, Canada	Lib., Aero. Lab.	1
National Center for Atmospheric Res.	Boulder, CO 80302	J. Curry	1
		J. Herring	1
National Science Foundation	Washington, DC 20550	Eng. Div.	1
Society of Naval Architects & Marine Engineers	New York, NY 10006		1

*Initial distribution of this document within the Applied Physics Laboratory has been made in accordance with a list on file in the APL Technical Publications Group.

INITIAL DISTRIBUTION EXTERNAL TO THE APPLIED PHYSICS LABORATORY*

ORGANIZATION	LOCATION	ATTENTION	No. cf Copies
MISCELLANEOUS (cont'd)			
Webb Institute of Naval Architecture	Glen Cove, NY 11542	Tech. Lib.	1
		E. Lewis	1
Applied Mechanics Review	San Antonio, TX 78206	Editor	1
Courant Inst. of Mathematical Sciences	New York, NY 10012	S. Childress	1
		E. Isaacson	1
		P. Lax	1
		L. Nirenberg	1
		J. Percus	1
		D. Stickler	1
Bassin d'Essais des Carenes	Paris, France	J. Dern	1
Oxford Univ. Computing Ctr.	Oxford, England	C. Elliott	1
		R. Furzeland	1
Ecole Polytechnique	Palaiseau Cedex, France	B. Mercier	1
		G. Nedelec	1
Inst.de Recherche d'Informatique et d'Automatique	Le Chesnay, France	O. Pironneau	1
H. Yuen	Redondo Beach, CA 90278		1

*Initial distribution of this document within the Applied Physics Laboratory has been made in accordance with a list on file in the APL Technical Publications Group.